



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

1990

Autopilot design for autonomous underwater vehicles based on sliding mode control

Lienard, David E.

Monterey, California: Naval Postgraduate School

<http://hdl.handle.net/10945/27778>

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

Downloaded from NPS Archive: Calhoun



Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

NAVAL POSTGRADUATE SCHOOL Monterey, California



THESIS

AUTOPILOT DESIGN FOR AUTONOMOUS
UNDERWATER VEHICLES
BASED ON SLIDING MODE CONTROL

by

David E. Lienard

June, 1990

Thesis Advisor:
Co-Advisor:

Fotis A. Papoulias
Anthony J. Healey

Approved for public release; distribution is unlimited.

Thesis
L63432
c.2

DODLEY HENRY LIBRARY
NAVAL COMMUNICATE SCHOOL
MCCLELLAN, CALIFORNIA 94033

REPORT DOCUMENTATION PAGE				
1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE				
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b. OFFICE SYMBOL (If applicable) ME	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School		
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS		
		Program Element No.	Project No.	Task No.
		Work Unit Accession Number		
11. TITLE (Include Security Classification) AUTOPILOT DESIGN FOR AUTONOMOUS UNDERWATER VEHICLES BASED ON SLIDING MODE CONTROL				
12. PERSONAL AUTHOR(S) Lienard, David E.				
13a. TYPE OF REPORT Mechanical Engineer	13b. TIME COVERED From To	14. DATE OF REPORT (year, month, day) June 1990	15. PAGE COUNT 130	
16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
17. COSATI CODES			18. SUBJECT TERMS (continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUBGROUP	Autonomous, Underwater Vehicles, AUV, Guidance, Control	
19. ABSTRACT (continue on reverse if necessary and identify by block number) An autopilot is designed using Sliding Mode Control Theory that can control the speed and heading of a full six degree of freedom, nonlinear model of the Navy's SDV Mark 9 vehicle. The control laws are based on a simplified linear model that allows speed and heading to be treated as separate systems. Once control of heading and speed is established, depth control is achieved by a third control law. Although they are developed separately, the three individual control laws act simultaneously to provide robust control of speed, heading, and depth of the nonlinear model of the vehicle. Line of Sight Guidance is used to convert the way points provided by the mission planner into commands for heading to which the autopilot responds. The performance of the autopilot is evaluated over a wide range of speeds to demonstrate its robustness. In addition, the effects of current are simulated and the autopilot is modified to compensate for the presence of a strong current.				
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS REPORT <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a. NAME OF RESPONSIBLE INDIVIDUAL Anthony J. Healey		22b. TELEPHONE (Include Area code) 408 646 2586	22c. OFFICE SYMBOL MEHy	

Approved for public release; distribution is unlimited.

Autopilot Design For Autonomous Underwater Vehicles
Based On Sliding Mode Control.

by

David E. Lienard
Lieutenant Commander, United States Navy
B.A., Illinois College
M.B.A., University of Central Florida

Submitted in partial fulfillment
of the requirements for the degree of

MASTERS OF SCIENCE IN MECHANICAL ENGINEERING
and
MECHANICAL ENGINEER

from the
NAVAL POSTGRADUATE SCHOOL
June 1990

Author:

David E. Lienard

Approved by:

Fotis A. Papoulias, Thesis Advisor

Anthony J. Healey, Co-Advisor

Anthony J. Healey, Chairman
Department of Mechanical Engineering

DEAN OF FACULTY
AND GRADUATE STUDIES

ABSTRACT

An autopilot is designed using Sliding Mode Control Theory that can control the speed and heading of a full six degree of freedom, nonlinear model of the Navy's SDV Mark 9 vehicle. The control laws are based on a simplified linear model that allows speed and heading to be treated as separate systems. Once control of heading and speed is established, depth control is then achieved by a third control law. Although they are developed separately, the three individual control laws act simultaneously to provide robust control of speed, heading, and depth of the nonlinear model of the vehicle. Line of Sight Guidance is used to convert the way points provided by the mission planner into commands for heading to which the autopilot responds. The performance of the autopilot is evaluated over a wide range of speeds to demonstrate its robustness. In addition, the effects of current are simulated and the autopilot is modified to compensate for the presence of a strong current.

Thesis
L63432
C-2

TABLE OF CONTENTS

I. INTRODUCTION	1
A. GENERAL	1
B. AIM OF THIS STUDY	2
C. THESIS OUTLINE	2
II. LINEAR MODELING WITH MATRIX-X	4
A. INTRODUCTION	4
B. AUV DYNAMIC MODEL	4
1. Identification of symbols	4
2. Simplification and linearization	6
a. Assumptions	6
b. Longitudinal equation of motion	6
c. Lateral equation of motion	7
d. Yaw equation of motion	8
e. Euler angle rate	8
f. Inertial position rates	8
3. State space configuration	9
a. SISO1	9

b. SISO2	9
C. SISO1 SLIDING MODE SPEED CONTROL	11
D. SISO2 SLIDING MODE HEADING CONTROL	14
E. SIMULATION RESULTS	19
III. CONTROLLING THE NONLINEAR AUV	24
A. INTRODUCTION	24
B. SPEED CONTROL [SISO1]	25
C. HEADING CONTROL [SISO2]	25
D. SIMULATION	26
E. CONCLUSION	33
IV. SWAY VELOCITY OBSERVER DESIGN	34
A. INTRODUCTION	34
B. OBSERVER DESIGN FOR LINEAR AUV	34
C. APPLICATION TO NONLINEAR AUV	42
D. EFFECT OF ASSUMING SWAY VELOCITY NEGLIGIBLE	45
E. CONCLUSION	46
V. GUIDANCE	49
A. INTRODUCTION	49
B. LINE OF SIGHT GUIDANCE	49

1. Fundamental Relationships	49
2. Interface parameters effecting performance	51
3. Results	53
VI. HIGH SPEED APPLICATIONS	60
A. INTRODUCTION	60
B. AUTOPILOT SIMULATION RESULTS	61
1. Run 1	61
2. Run 2	62
3. Run 3	62
4. Conclusion	69
C. LOS GUIDANCE SIMULATIONS	69
1. Run 4	69
2. Run 5	70
3. Run 6	70
4. Conclusion	73
VII. EFFECT OF CURRENT ON LOS GUIDANCE	75
A. INTRODUCTION	75
B. EFFECT OF CURRENT DIRECTION	75
1. Run 1	75
2. Run 2 and Run 3	77

3. Run 4 and Run 5	77
4. Conclusion	77
C. ADAPTATION OF STEERING CONTROL	80
D. MULTIPLE WAY POINT PATHS	83
VIII. CONCLUSIONS AND RECOMMENDATIONS	91
A. CONCLUSIONS	91
B. RECOMMENDATIONS	92
APPENDIX A	93
APPENDIX B.	96
APPENDIX C.	107
APPENDIX D.	112
LIST OF REFERENCES	116
INITIAL DISTRIBUTION LIST	118

ACKNOWLEDGMENTS

I would like to thank my thesis advisor, Professor Fotis A. Papoulias and co-advisor, Professor Anthony J. Healey for all their help and encouragement. Their ability to create a team atmosphere for the many students and faculty that are interested in developing an increased understanding of autonomous underwater vehicles was the key element in making this thesis an enjoyable and rewarding experience. I would also like to thank Joo-No Sur for his fine work that laid a solid foundation for me to build upon and David Marco for his advice and support regarding the Vax computer system.

In addition, I would like to acknowledge the financial support for the NPS Autonomous Underwater Vehicle Project from the NPS Directed Research Fund with the technical sponsorship of the Naval Surface Warfare Center.

Finally, I would like to acknowledge the support of my wife and family without which I would not have been free to devote the time and thought required to take advantage of the opportunity that my assignment to NPS provided.

I. INTRODUCTION

A. GENERAL

There has been an increased interest recently in the need for autonomous underwater vehicles (AUV) in both the Navy and private industry. A variety of unclassified missions include ASW, decoy, survey, reconnaissance, and ocean engineering work. As the cost of manned submarines increases, there are significant advantages to using cheaper unmanned vehicles. The AUV should be able to maneuver freely in the ocean environment with respect to speed, heading, and depth in order to carry out its missions. Such maneuvering requirements have to be easily accomplished by a low level active control system, and in the presence of environmental and physical uncertainty.

All information concerning the environment of a vehicle is detected by the sensing instrumentation on board the vehicle and sent to the higher level intelligence systems to be interpreted in order to carry out the mission. The dynamics of underwater vehicles are described by highly nonlinear systems of equations with uncertain coefficients and disturbances that are difficult to measure. Robust control using Sliding Mode Control Theory is reputed to provide accurate control of nonlinear systems despite unmodeled system dynamics and disturbances making it a likely candidate for designing the control laws that will govern the autopilot function of unmanned vehicles.

B. AIM OF THIS STUDY

The aim of this thesis is to develop a robust autopilot that will control speed, heading, and depth by the application of Sliding Mode Control Theory. The full six degree of freedom equations of motion that describe the vehicle are simplified and linearized to the point that they can be treated as three separate, independent, single input single output systems. The control laws for speed, heading, and depth are developed independently and then combined together to attain effective control of a simple linear model first and ultimately the complete nonlinear model for the vehicle.

Once the autopilot is verified, it is coupled with a Line of Sight Guidance scheme to allow the autopilot to interface with the higher levels of decision making that provide way points to describe a desired path for the vehicle. The merits of the total system are then evaluated for robustness in terms of the range of operating speed over which it can successfully control the vehicle when compared to the nominal operating speed about which the equations of motion were initially linearized.

Finally, the effect of current on Line of Sight Guidance is explored and refinements made resulting in the development of a highly robust and effective system for controlling the next generation of autonomous underwater vehicles under construction at the Naval Postgraduate School and elsewhere in private industry.

C. THESIS OUTLINE

In Chapter 2 the equations of motion for the full six degree of freedom model of the AUV are simplified and linearized. A model for the vehicle is then built using the

Matrix-x software package (Copyright 1989 by Integrated Systems Inc.) and sliding mode control laws are developed, assuming full state feedback, to control the speed and heading of this simple linear model.

Chapter 3 shows the application of the control laws developed in Chapter 2 to the full, nonlinear, six degree of freedom model that represents the SDV-9 vehicle currently in use by the Navy. In addition, the depth control law which was developed previously at NPS by Joo-No Sur [Ref. 1] was included to attain control of depth.

In Chapter 4 an observer for sway velocity is designed and tested. It is ultimately determined that sway velocity is better treated as a disturbance and consequently it is set to zero wherever it appears in the control law equations.

Chapter 5 explores using Line of Sight Guidance to allow the way points provided by the planning function of the AUV to be interpreted into commands for heading.

Chapter 6 investigates the speed envelope of 3 ft/sec to 50 ft/sec over which the control laws are valid even though the initial linearization was done around an operating speed of 6 ft/sec. This shows the robustness of the control laws.

Chapter 7 shows the effect of current on Line of Sight Guidance and develops a modified control law, including current compensation, that improves the autopilot's effectiveness in currents with speeds up to 50% of the operating speed.

II. LINEAR MODELING WITH MATRIX-X

A. INTRODUCTION

Because the vehicle dynamic equations of motion are relatively easy to linearize, the first step of control design for the autopilot was to consider the AUV to be represented by a nominal linear model. In linearizing the equations of motion it was found that the equations became uncoupled to the extent that speed control and heading control could be addressed separately. This led to considering the control of heading and speed as two independent SISO systems. Using this design approach, the two inputs provided to the autopilot are the desired heading and the desired speed. No attempt is made to control depth at this stage in the autopilot design. In fact, it is assumed that all motion is confined to the horizontal plane. The block diagram representing the autopilot is shown in Figure 1. This is the actual Matrix-x system build diagram in superblock format. The remaining portion of this chapter will be devoted to explaining each of these blocks individually as the autopilot control design, based on the nominal model, is described.

B. AUV DYNAMIC MODEL

1. Identification of symbols

All the symbols that will be used to develop the linear model for the AUV are identified in the execution program shown in Appendix A. They are listed along with their units and in the case of constant parameters, numerical values are given.

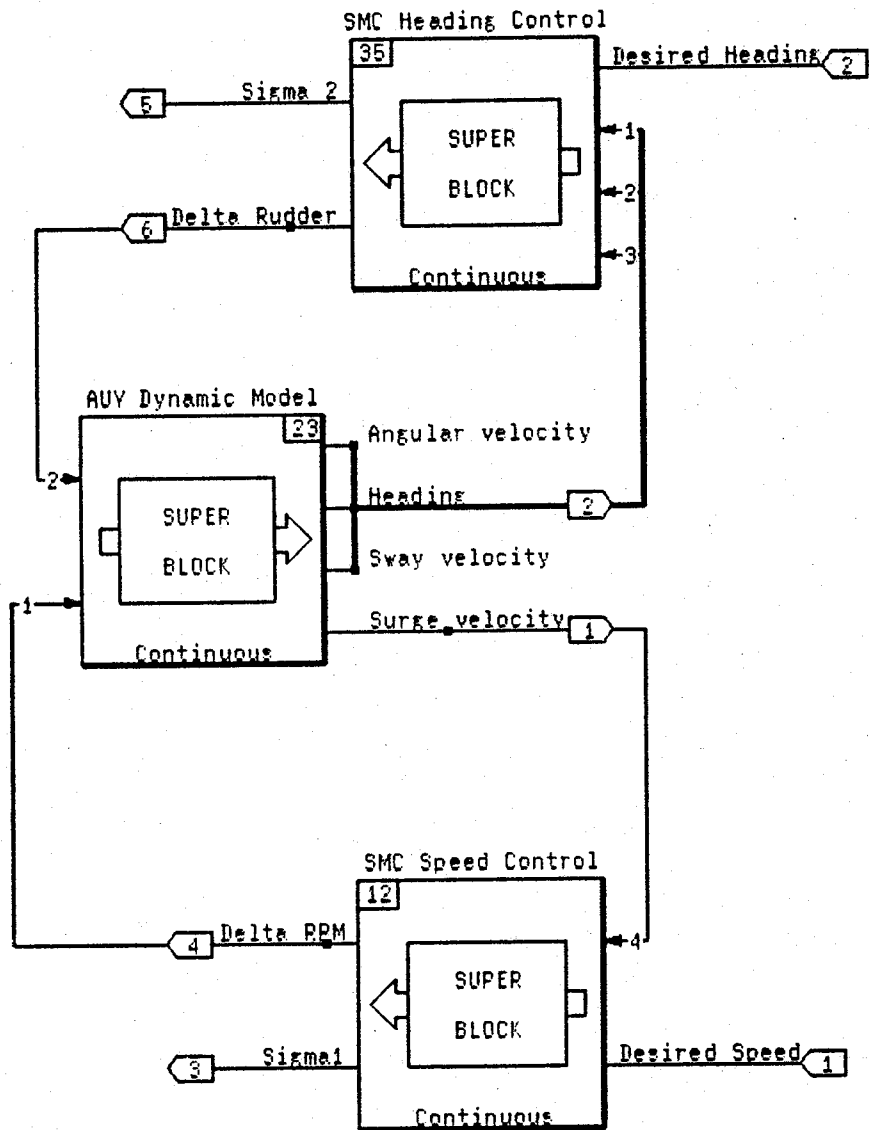


Figure 1. Autopilot Block Diagram

2. Simplification and linearization

The nonlinear equations of motion for this vehicle were taken from work done previously at NPS by Boncal [Ref. 2] who used the dynamic model as established by Crane, Summey, et al [Ref. 3], as representative of the SDV Mark 9 vehicle. While this vehicle is not identical to any specifically planned AUV, it remains a useful vehicle for the study of dynamics and control issues.

Only the equations for motion in the horizontal plane are developed in this chapter. In addition, the complete form of these equations will not be shown. Instead, the assumptions used to simplify them will be stated and the resulting simplified equations will be shown in their linearized form. It should be noted that the linearization was done around a constant speed $\bar{u} = 6.0$ ft/sec and a corresponding constant motor rpm $\bar{n} = 500$ rpm, with the vehicle in level flight.

a. Assumptions

- Motion is confined to the horizontal plane:

$$(w, p, q, \theta, \phi, \dot{p}, \dot{q}, \delta_b, \delta_s = 0)$$

- Some dynamic coefficients are small and can be neglected:

$$(Y_p, Y_{\dot{p}}, Y_{pq}, Y_{qr} = 0)$$

- The AUV is neutrally buoyant.

- By design X_g, Y_g, Z_g are negligible.
- Cross flow drag terms tend to cancel each other as far as steering moments are concerned.

b. Longitudinal equation of motion

$$\dot{u} = -\alpha \bar{u} u + \beta \bar{n} \delta_n \quad (2.1)$$

Where:

$$\alpha = (2 \rho L^2 C_{D0}) / (2 \text{ mass} - \rho L^3 X_u)$$

$$C_{D0} = .00385 + 1.296E-17 [\bar{u} L/kvis - 1.2E-7]^2$$

$$\beta = (.012)^2 \alpha$$

c. Lateral equation of motion

$$m_1 \dot{v} - m_2 \dot{r} = Y_1 \bar{u} v + Y_2 \bar{u} r + Y_3 \delta_r \quad (2.2)$$

Where:

$$m_1 = (\text{mass} - \rho L^3 Y_v) / 2$$

$$m_2 = (\rho L^4 Y_r) / 2$$

$$Y_1 = (\rho L^2 Y_v \bar{u}) / 2$$

$$Y_2 = (\rho L^3 Y_r \bar{u}) / 2$$

$$Y_3 = (\rho L^2 Y_{\delta_r} \bar{u}^2) / 2$$

d. Yaw equation of motion

$$m_3 \dot{v} + m_4 \dot{r} = N_1 v + N_2 r + N_3 \delta_r \quad (2.3)$$

Where:

$$m_3 = (-\rho L^4 N_v) / 2$$

$$m_4 = I_z - (\rho L^5 N_r) / 2$$

$$N_1 = (\rho L^3 N_v \bar{u}) / 2$$

$$N_2 = (\rho L^4 N_r \bar{u}) / 2$$

$$N_3 = (\rho L^3 N_{\delta_r} \bar{u}^2) / 2$$

e. Euler angle rate

$$\dot{\psi} = r \quad (2.4)$$

f. Inertial position rates

$$\dot{x} = u \cos(\psi) - v \sin(\psi) \quad (2.5)$$

$$\dot{y} = u \sin(\psi) + v \cos(\psi) \quad (2.6)$$

3. State space configuration

a. *SISO1*

The state space equation for system 1 that will be used to attain speed control is simply the longitudinal equation of motion previously developed and considered to be in the standard state space form shown below.

$$\dot{X}_1 = [A_1] X_1 + [B_1] U_1$$

The components of these matrices are given by equation (2.7).

$$\dot{u} = [-\alpha \bar{u}] u + [\beta \bar{n}] \delta_n \quad (2.7)$$

b. *SISO2*

To find the state space equation for heading control, equations (2.2) and (2.3) must be treated as coupled. This was done as follows:

$$M = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix}$$

$$C = \begin{bmatrix} Y_1 & Y_2 & Y_3 \\ N_1 & N_2 & N_3 \end{bmatrix}$$

Represent equations (2.2) and (2.3) as:

$$[M] \begin{pmatrix} \dot{v} \\ \dot{r} \end{pmatrix} = [C] \begin{pmatrix} v \\ r \\ \delta_r \end{pmatrix}$$

Consequently, this equation can be solved as follows:

$$\begin{pmatrix} \dot{v} \\ \dot{r} \end{pmatrix} = [M]^{-1} [C] \begin{pmatrix} v \\ r \\ \delta_r \end{pmatrix} = [D] \begin{pmatrix} v \\ r \\ \delta_r \end{pmatrix} \quad (2.8)$$

Combining equation (2.4) with equation (2.8) results in the state space form to be used for heading control. Since two systems are being dealt with at the same time this system will be represented by the following state space equation.

$$\dot{X}_2 = [A_2] X_2 + [B_2] U_2$$

Where these matrices are defined by equation (2.9).

$$\begin{pmatrix} \dot{\psi} \\ \dot{v} \\ \dot{r} \end{pmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & d_{11} & d_{12} \\ 0 & d_{21} & d_{22} \end{bmatrix} \begin{pmatrix} \psi \\ v \\ r \end{pmatrix} + \begin{bmatrix} 0 \\ d_{13} \\ d_{23} \end{bmatrix} \delta_r \quad (2.9)$$

Equations (2.7) and (2.9) are represented graphically in the Matrix-x system build diagram shown in Figure 2. Again notice that they represent two independent

systems which leads to the next step of determining the control laws for heading control and speed control by considering them as separate SISO systems. The triangles represent gain values calculated in the execution program shown in Appendix A.

C. SISO1 SLIDING MODE SPEED CONTROL

At this point the control of speed is easily achieved using sliding mode control theory. The basis of this theory and its application to controlling the dive maneuvers of AUV's is documented by Joo-No Sur [Ref. 1]. Consequently, the theory of sliding mode control will not be included here but rather its application will be shown. Further details are given in the tutorial by Decarlo et al. [Ref. 4], and in the paper by Yoeger and Slotine [Ref. 5] for application to underwater vehicles.

Recalling from equation (2.7) that the state vector for this system is simply the surge velocity (u) and the input is the motor rpm (δ_m), the equation for the sliding surface is written as a weighted sum of the error between the state variables and their desired values.

$$\sigma_1 = s_1(u - u_d) \quad (2.10)$$

According to Liapunov Stability [Ref. 6], global asymptotic stability of the speed error is guaranteed by the condition:

$$\dot{\sigma}_1(t) \sigma_1(t) < 0$$

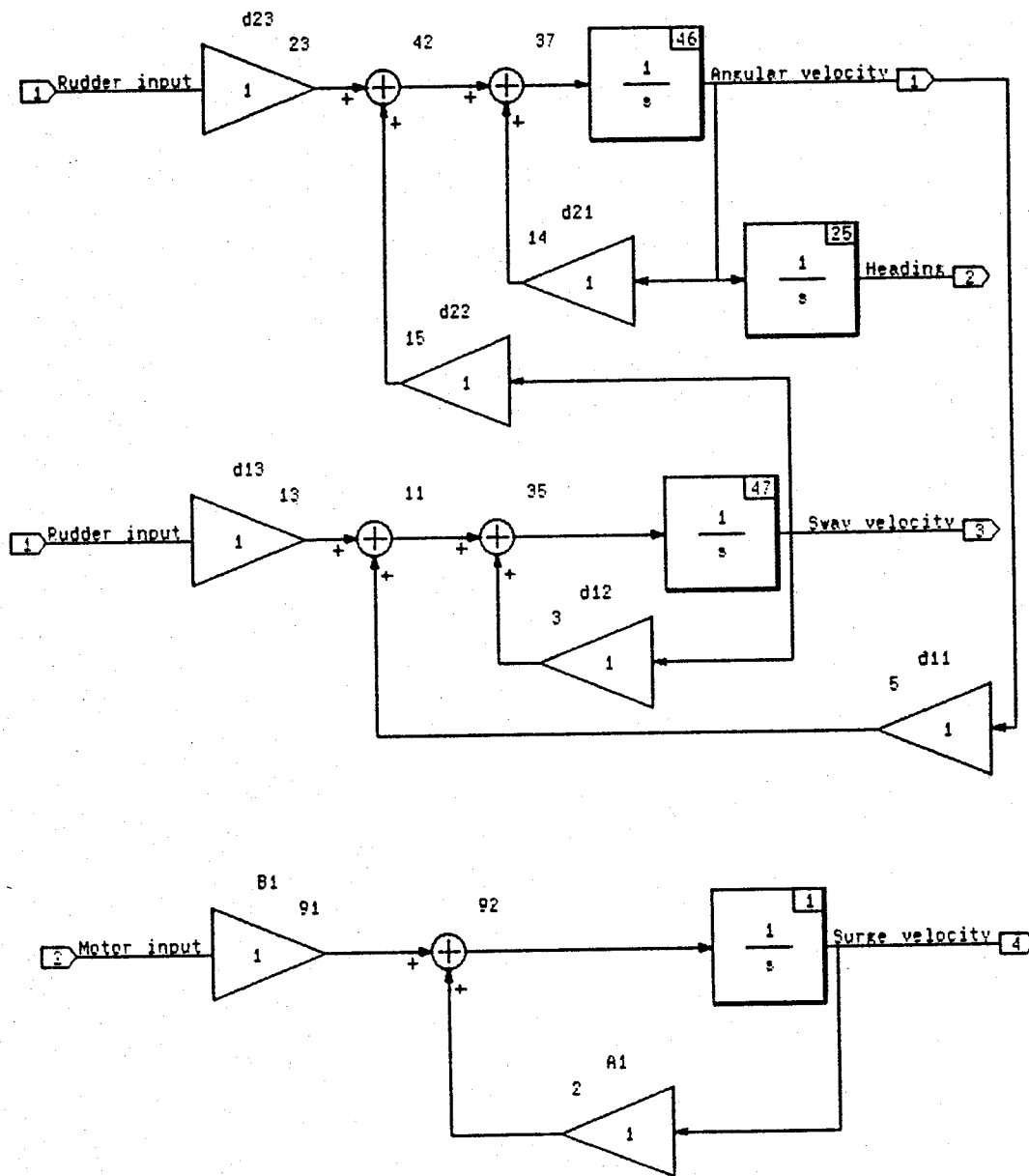


Figure 2. Linear AUV Dynamic Model

For SISO1 this results in the following equation.

$$\dot{\sigma}_1(t) = -(\eta_1^2) \text{sgn}(\sigma_1(t)) \quad (2.11)$$

The symbol η_1 represents the sliding coefficient for the speed control law. To eliminate chattering, the **sgn** function can be replaced without loss of generality by the Matrix-x function **satsgn**. **Satsgn** still acts like a switch, but has the advantage of allowing a linear increase to the prescribed saturation value which in this case is unity. As a result of this substitution, and considering the time dependence of σ_1 to be understood, the expression now has the form:

$$\dot{\sigma}_1 = -(\eta_1^2) \text{satsgn}(\sigma_1) \quad (2.12)$$

The speed control law can be obtained by setting the derivative of Equation (2.10) equal to Equation (2.12) and substituting for \mathbf{u} from the state space equation for SISO1. The resulting expression for the motor input is:

$$\delta_n = [-\eta_1^2 \text{satsgn}(\sigma_1) - s_1 \alpha \bar{u}] / (s_1 \beta \bar{n}) \quad (2.13)$$

At this point the values of η_1 and s_1 must be determined. Since there is only one state variable to consider, $s_1 = 1$. The choice of η_1 will be made through trial and error using Matrix-x simulation. In general, increasing η_1 will make the system more responsive while reducing it will make the system more sluggish.

The speed control law is represented in Matrix-x system build in Figure 3. Notice that the control signal (δ_n) is made up of two parts, one representing the linear feedback term and the other the nonlinear switching term as shown as follows:

$$\delta_n = \hat{\delta}_n + \bar{\delta}_n \quad (2.14)$$

Notice too, that an upper limit of 500 rpm has been included in the system build to simulate the maximum allowed motor rpm.

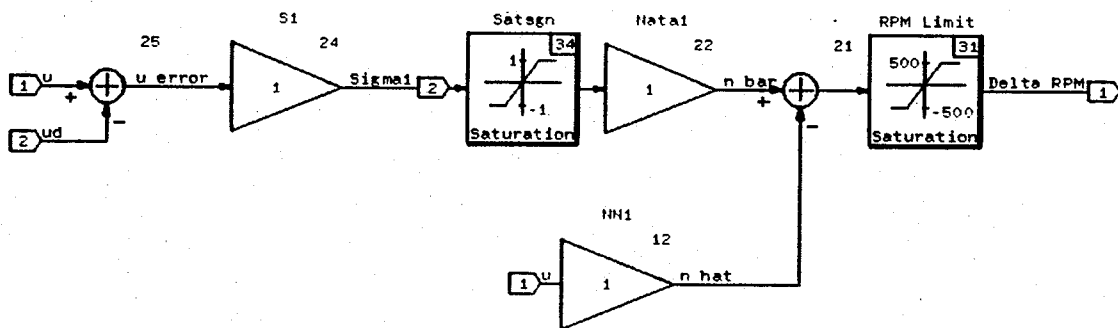


Figure 3. SISO1 Sliding Mode Speed Control Block Diagram

D. SISO2 SLIDING MODE HEADING CONTROL

Sliding mode control theory is used to achieve heading control by using the state space equation for SISO2. First the equation for the sliding surface is written as a

weighted sum of the error between the state variables and their desired values where the weights are the coefficients of the sliding surface [Ref. 4].

$$\sigma_2 = s_2(\psi - \psi_d) + s_3(v - v_d) + s_4(r - r_d) \quad (2.15)$$

This equation is simplified by noting that the desired values of sway velocity and angular velocity are zero. By making this substitution the sliding surface equation for heading control becomes:

$$\sigma_2 = s_2(\psi - \psi_d) + s_3(v) + s_4(r) \quad (2.16)$$

Again using Liapunov Stability Theory to meet the requirements for global asymptotic stability the following expression is obtained for SISO2.

$$\dot{\sigma}_2 = -(\eta_2)^2 \text{sat} \text{sgn}(\sigma_2) \quad (2.17)$$

Next, $\dot{\sigma}_2$ is found by taking the derivative of equation (2.16), realizing that ψ_d , at this stage, is considered to be a constant.

$$\dot{\sigma}_2 = s_2(\dot{\psi}) + s_3(\dot{v}) + s_4(\dot{r}) \quad (2.18)$$

By equating equation (2.17) and (2.18) and substituting from the SISO2 state space equation for the derivatives of the state variables the heading control law is obtained as shown.

$$\delta_r = \frac{[-(\eta_2)^2 \text{satsgn}(\sigma_2) - (s_2 + s_3 d_{11} + s_4 d_{21})r]}{(s_3 d_{13} + s_4 d_{23})} \quad (2.19)$$

Notice again, that this control law can again be considered to have two distinct parts as shown below.

$$\delta_r = \hat{\delta}_r + \bar{\delta}_r$$

The linear feedback part of the control law is designed to give the system the desired dynamics while on the sliding surface. The nonlinear term is responsible for getting the system to the sliding surface initially and compensating for any unmodeled behavior that would drive the vehicle off of the sliding surface once it got there. Once on the sliding surface, there is no need for the nonlinear term unless there is uncertainty in the model [Ref. 4]. The equations for both of these terms are as follows:

$$\hat{\delta}_r = -[NN_1 r + NN_2 v] / DD_2 \quad (2.20)$$

$$\bar{\delta}_r = (-\eta_2)^2 / DD_2 \text{satsgn}(\sigma_2) \quad (2.21)$$

Where:

$$NN_1 = s_2 + s_3 d_{11} + s_4 d_{21}$$

$$NN_2 = s_3 d_{12} + s_4 d_{22}$$

$$DD_2 = s_3 d_{13} + s_4 d_{23}$$

Unlike the speed control subsystem, the sliding surface for heading control is multidimensional. In this case the value for $S^T = [s_2, s_3, s_4]^T$ is determined by considering the closed loop dynamics of this system in the sliding condition ($\dot{\sigma}, \sigma = 0$). Under this condition the state space equation for SISO2 becomes:

$$\dot{X}_2 = [A_2] X_2 + [B_2] \delta_r \quad (2.22)$$

By writing equation (2.20) in terms of S^T and the state space matrices, the following relationship is attained.

$$\delta_r = -(S^T B_2)^{-1} S^T A_2 X_2 \quad (2.23)$$

Combining equations (2.22) and (2.23) gives:

$$\dot{X}_2 = [A_2 - B_2(S^T B_2)^{-1} S^T A_2] X_2 \quad (2.24)$$

Note that equation (2.24) is in the form:

$$\dot{X}_2 = [A_2 - B_2 K] X_2$$

Where:

$$K = (S^T B_2)^{-1} S^T A_2$$

Consequently, equation (2.24) can be represented as:

$$\dot{X}_2 = [AC_2] X_2$$

In this form $[AC_2]$ is the closed loop dynamics matrix. At this point K can be found using pole placement techniques. This is done by specifying the desired poles that will determine the behavior on the sliding plane. It should be noted here that in accordance with Sliding Mode Theory, one of these poles must be at the origin thus leaving 2 poles to determine the closed loop bandwidth and damping [Ref. 4]. Using Matrix-x this is done as follows:

$$K = \text{POLEPLACE}(AC_2, B_2, \text{POLES}_2)$$

Now that a value for K can be found, S^T is obtained by multiplying both sides of the equation for K by $S^T B_2$. This gives the result:

$$S^T [AC_2] = 0$$

Which is also correctly expressed as equation (2.25).

$$[AC_2]^T S = 0 \tag{2.25}$$

At this point the fact that $[AC_2]^T$ has one eigenvalue that is zero is utilized to complete the solution of S by considering the standard eigenvalue problem $([AC_2]^T - I)V = 0$. For the case when the eigenvalue is zero, $[AC_2]^T V = 0$. Comparing this expression

with equation (2.25) it becomes apparent that the eigenvector V in the general case becomes the vector S for our circumstances [Ref. 7]. Consequently, S is determined by the eigenvalue function available in Matrix-x. This calculation is performed within the execution program shown in Appendix A.

The next step is to test the control law via simulation by adjusting the values of η_2 and the poles to obtain the desired control characteristics. The Matrix-x system build representation of the heading controller that was used for simulation is shown in Figure 4. Notice that the **satsgn** function has been modified. Its slope has been adjusted to provide a filter for numerical chattering. The saturation limit of the **satsgn** function represents the boundary layer thickness ϕ as developed by Slotine [Ref. 5]. Since the range of values of the **satsgn** function is between -1 and +1, a multiplication factor of $1/\phi$ must also be included.

Selection of the boundary layer thickness is a design issue. In Figure 4, the value is set at .1 based on many simulation runs. This in turn requires a multiplication factor of 10 to be included in the block diagram.

E. SIMULATION RESULTS

The goal for the controller was to reach the desired heading and speed in the shortest time possible and avoid inefficient use of the control inputs while doing so. Many simulation runs were conducted to obtain the best values for η_1 , η_2 , and the poles for the dynamics of the sliding surface of SISO2. In addition, reasonable values for the

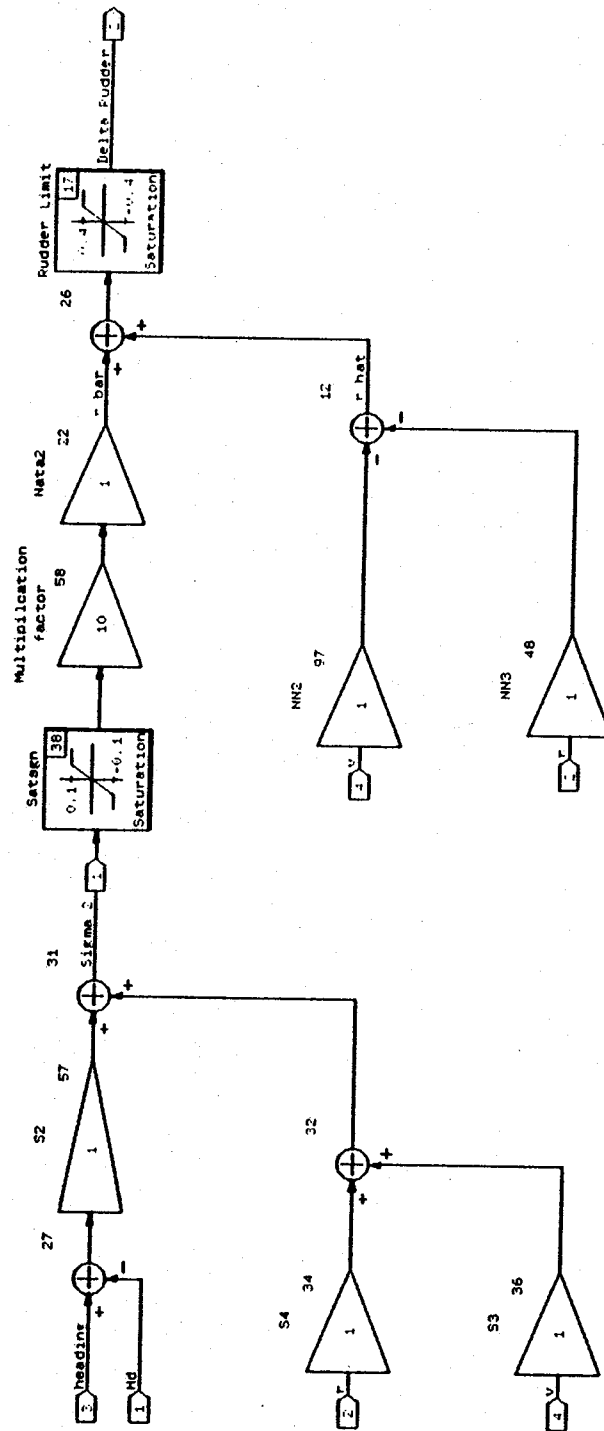


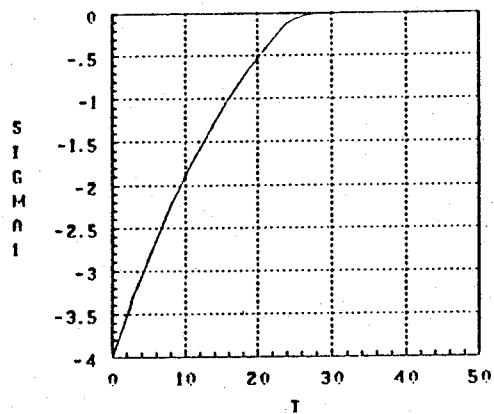
Figure 4. SISO2 Sliding Mode Heading Control Block Diagram

boundary layer thicknesses and for the **satsgn** functions had to be determined.

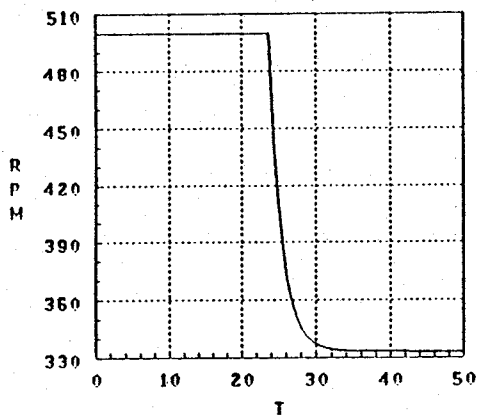
The output from the simulation is displayed in a set of six graphs. The graph of Sliding Coefficient vs Time gives insight into the role that the η and the choice of poles play in minimizing the amount of time it takes to reach the desired heading. The value of η primarily affects the time it takes to reach the sliding plane where $\sigma(x) = 0$. The poles that are selected also effect the time it takes to reach the sliding plane by effecting its slope but their primary influence is on the dynamics of the system as it travels along the sliding plane to reach the desired speed and heading. This shows the need to balance the value of η and the poles to obtain the optimal behavior for the control system. It appears as though a threshold for η exists above which the time to reach the sliding plane no longer decreases and the only advantage that is gained by increasing η would be to increase the robustness of the controller as it tries to maintain the system on the sliding plane in the presence of unmodeled dynamics and disturbances. Recalling that no uncertainties have been built into this model yet, the choice of the best η values is based on response time alone. Initial runs caused a value of .6 to be selected for η_1^2 and .4 for η_2^2 . Using these values additional simulations revealed the best choice of poles for SISO2. Heading control was attained with a pole selection of [0,.41,.42]. A more detailed description of the effects of each of these elements of a sliding mode controller along with simulation runs to reveal their effects appears in [Ref. 1].

This control law was used for the simulation run shown in Figure 5. The AUV was initially at rest and commanded to attain a constant speed of 4 ft/sec. While the speed controller was accomplishing that task the heading controller was independently changing

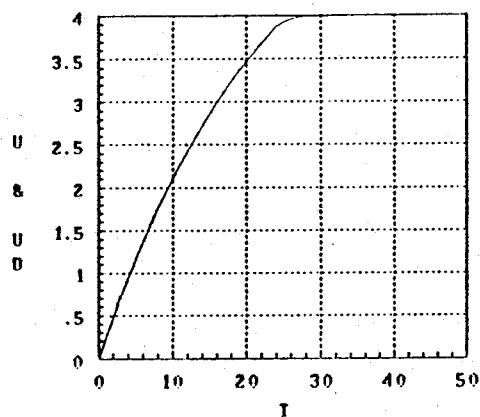
the heading to 90 degrees from an initial value of zero. The results show a good response with efficient use of the control inputs. These results are typical of the many trials that were run to verify the control laws. The next step is to apply these control laws based on a simplified, linear model, to a more realistic nonlinear model to see if they can still be effective in controlling speed and heading.



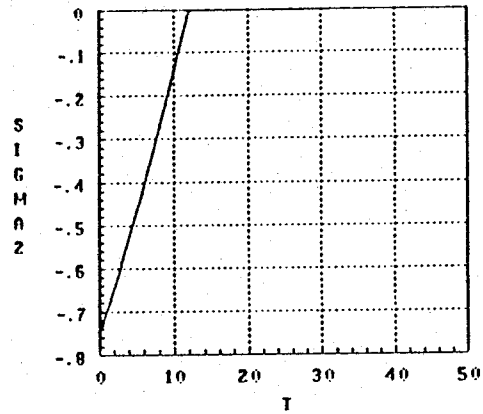
ATASQ1=.6 BAND=1FPS



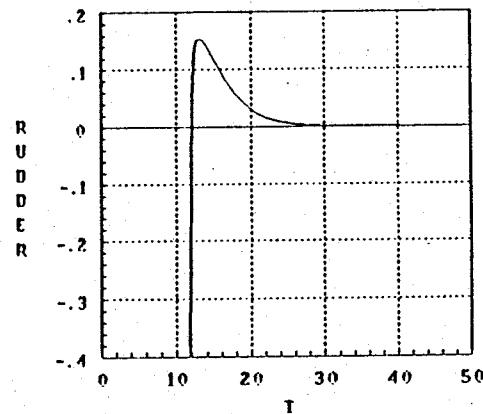
PROPULSION (MAX RPM = 500RPM)



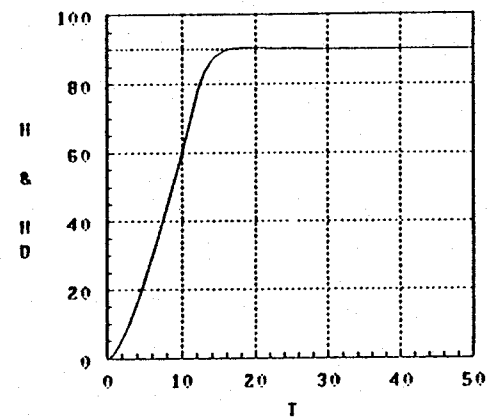
SPEED [$U_d = 4$ FPS]



ATASQ2=.4 POLES=[0,-.4,-.42] BAND=.1RAD.



STEERING (MAX DEFLECTION=.4RAD)



HEADING [$H_d = 90$ DEG]

Figure 5. Simulation Results for Linear AUV

III. CONTROLLING THE NONLINEAR AUV

A. INTRODUCTION

Two main questions are addressed in this chapter. First, will the sliding mode control laws for speed and heading developed in Chapter 2 using a simplified linear model for the AUV be able to control a more realistic nonlinear model of the AUV. The nonlinear model that is used was originally developed by Boncal [Ref. 2], based on the Mk 9 swimmer delivery vehicle and most recently used by Joo-No Sur [Ref. 1] in his design of a SMC depth controller. It is in the form of a Fortran code and is shown in Appendix B.

The second question to be answered is whether the control laws for speed and heading control can be simply inserted into the Fortran model as two independent modules and work harmoniously with the depth controller already present. If this is possible then the task of three dimensional control of the nonlinear six degree of freedom model can be accomplished by 3 simple SISO systems rather than a complicated MIMO system. Recalling that many simplifying assumptions were made to develop the linear model that are no longer valid for the nonlinear model and that some of the assumptions for one of the SISO systems may be in direct conflict with the others, it is likely that significant unmodeled behavior will have to be handled by the nonlinear switching terms in the SMC control laws.

B. SPEED CONTROL [SISO1]

Recall from Chapter 2 that the best value of η_1^2 was .6. Utilizing Matrix-x to get the necessary coefficients, the speed control law was obtained and programmed into the Fortran code as shown in Figure 6. Notice the use of IF THEN statements to provide the satsgn function and to establish an RPM limit.

```
*****RPM INPUT CALCULATION *****  
SS1=U-UD  
IF(ABS(SS1) .LT. 1.0) SATSGN1=(SS1/1)  
IF(SS1 .LE. -1.0) SATSGN1=-1.0  
IF(SS1 .GE. 1.0) SATSGN1=1.0  
RPM=-1153.9*SATSGN1 + 83.33*U  
IF (RPM .GE. 500.0) RPM= 500.0  
IF (RPM .LE. -500.0) RPM=-500.0
```

Figure 6. Sliding Mode Speed Control Fortran Code

C. HEADING CONTROL [SISO2]

The Matrix-x simulations provided the necessary values for the control law. Recall that a value of .4 was chosen for η_2^2 and that the pole location of [0, -.41, -.42] was selected. This resulted in the following heading control law shown in Figure 7 that was programmed into the Fortran model. Notice that the maximum rudder deflection has been limited to $\pm .4$ radians.

```

*****RUDDER INPUT CALCULATION*****
SS2=.5732*(PSI-HD) - .0739*V + .8161*R
IF(ABS(SS2) .LT. .1) SATSGN2=(SS2/.10)
IF(SS2 .LE. -.1) SATSGN2=-1.0
IF(SS2 .GE. .1) SATSGN2=1.0

DR=2.5785*SATSGN2 + .0328*V + .1112*R
IF (DR .GE. 0.4) DR = 0.4
IF (DR .LE. - 0.4) DR = -0.4

```

Figure 7. Sliding Mode Heading Control Fortran Code

D. SIMULATION

With these two primary modifications to the Fortran model, the same maneuver that was shown in Chapter 2 using the linear model was now attempted for the nonlinear AUV. The control laws were very successful at achieving the desired speed and heading as can be seen by Figure 8. The effects of the unmodeled terms were present, but stabilized by the nonlinear switching term in the control laws confirming the robust nature of sliding mode control that make it particularly desirable for this application. The response was more sluggish and more control effort was required than for the linear model. This was expected however since the nonlinear model contains terms to represent the increased loss in a turn which the linear model did not. A comparison to Figure 5, in Chapter 2, shows that it took about 15 seconds longer to reach the desired speed and almost 30 seconds longer to reach the desired heading. These results seem very

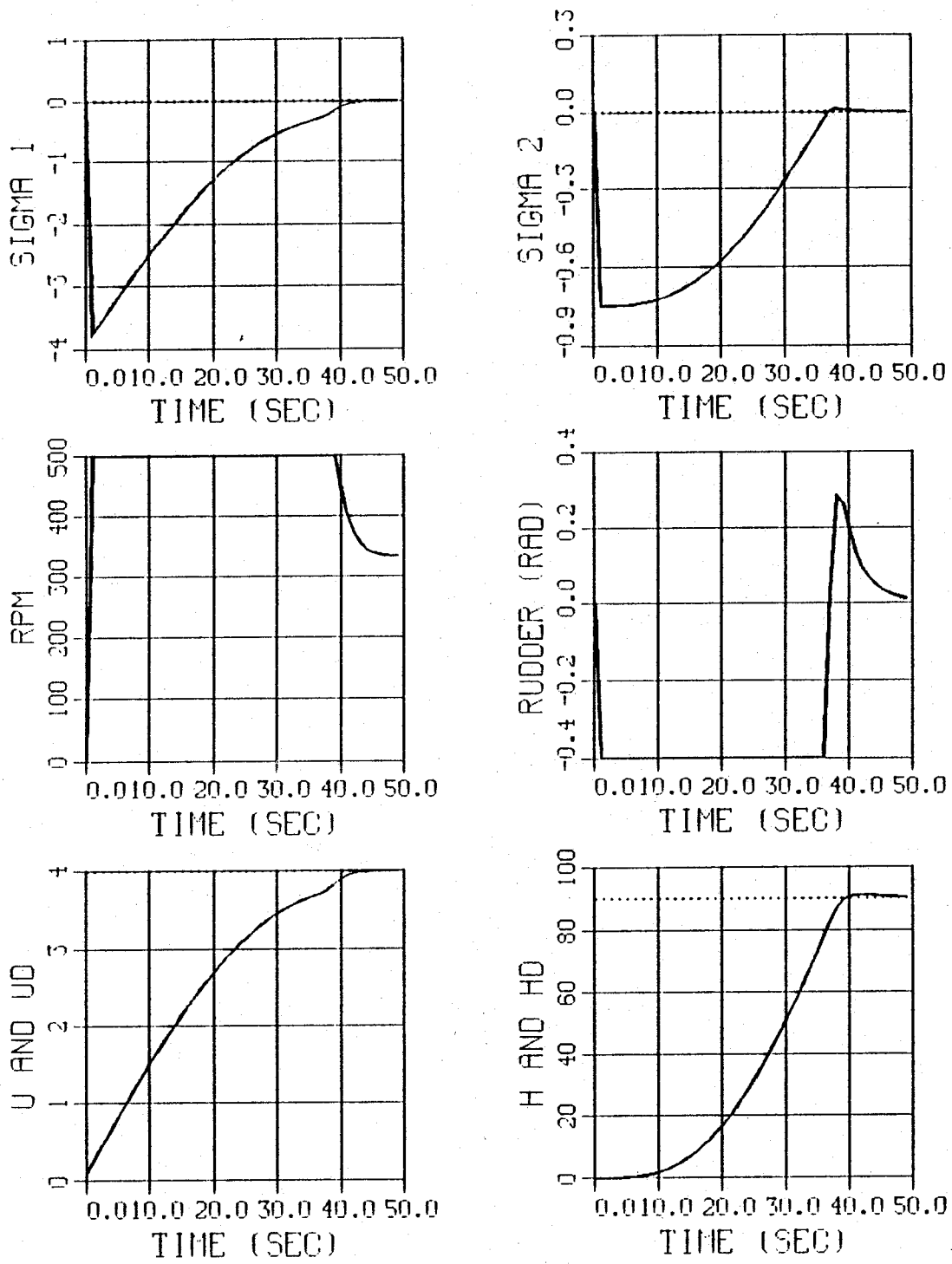


Figure 8. Simulation Results for Nonlinear AUV

satisfactory since the use of the motor and rudder inputs were at their maximum values throughout most of the run and consequently utilized to their full capacity to reach the desired conditions.

To answer the second question the same maneuver was commanded with a depth change of 20 feet included. It should again be pointed out that the depth control law was developed independently by Joo-No Sur and used without any modification [Ref. 1]. The results are shown in Figures 9 and 10. Notice that the depth control law accomplished the commanded depth change independent of the course and speed adjustments. The control effort of the dive planes seems very efficient as does the use of the rudder and motor RPM. All three of these control laws are acting simultaneously and independently but together they produce the desired result.

To further demonstrate the independence of these 3 SISO systems, two additional runs are shown. First, the 20 foot depth change is commanded without requiring a change in heading. The depth change characteristics are then compared to those for the previous simulation in Figure 11. The left column is the run with a depth change only and the right column shows the depth change while the vehicle is also making a 90 degree heading correction. Notice that the results are almost identical showing that depth control is not adversely affected by a change in heading.

Similarly, a 90 degree turn was commanded without requiring a depth change and compared to the combined diving and turning maneuver. The results are compared in Figure 12. The left column represents the turn only while the right column shows the same turn during a 20 foot depth change. Again the results are nearly identical showing

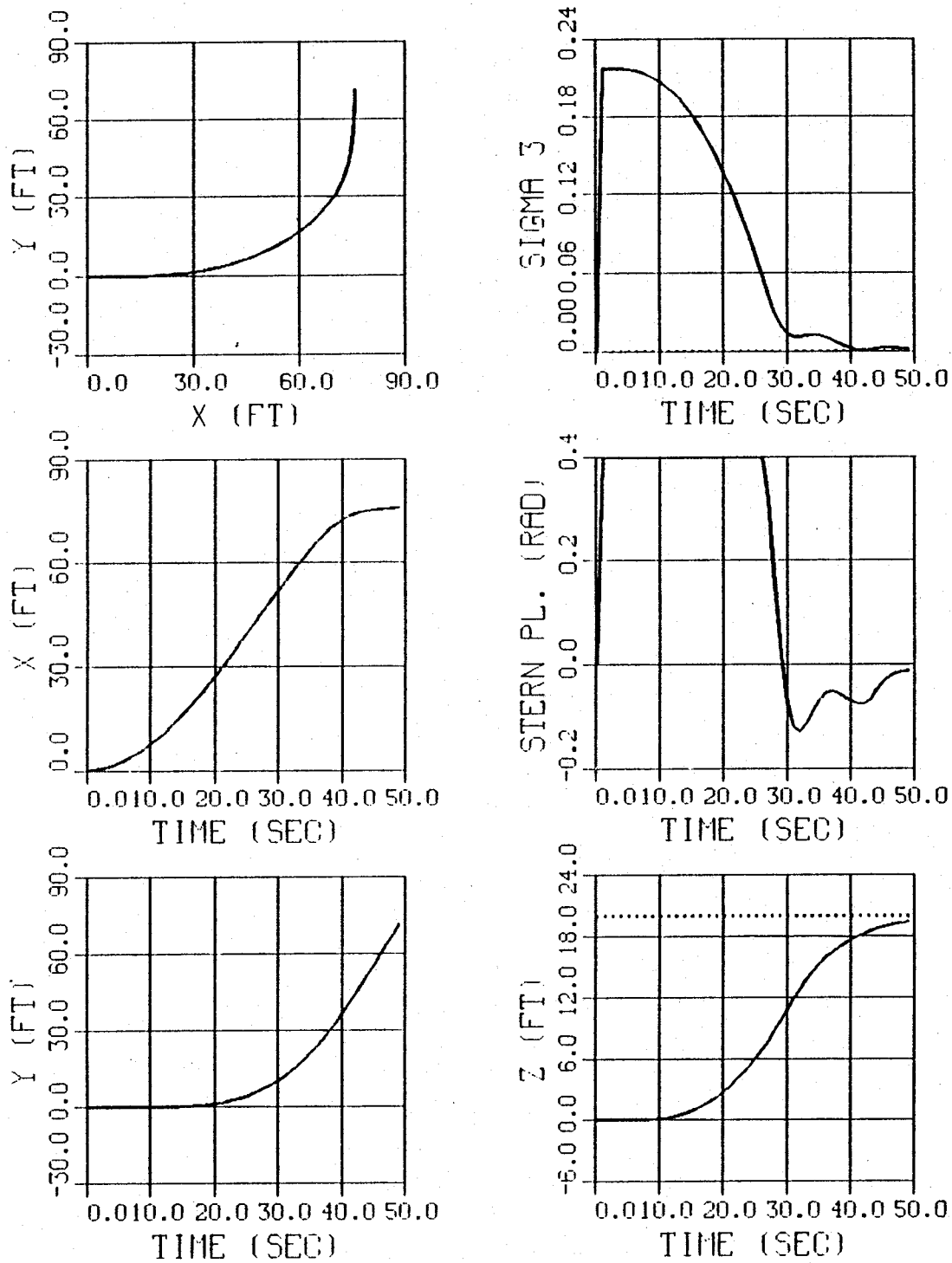


Figure 9. Simulation Including Depth Change (Nonlinear AUV)

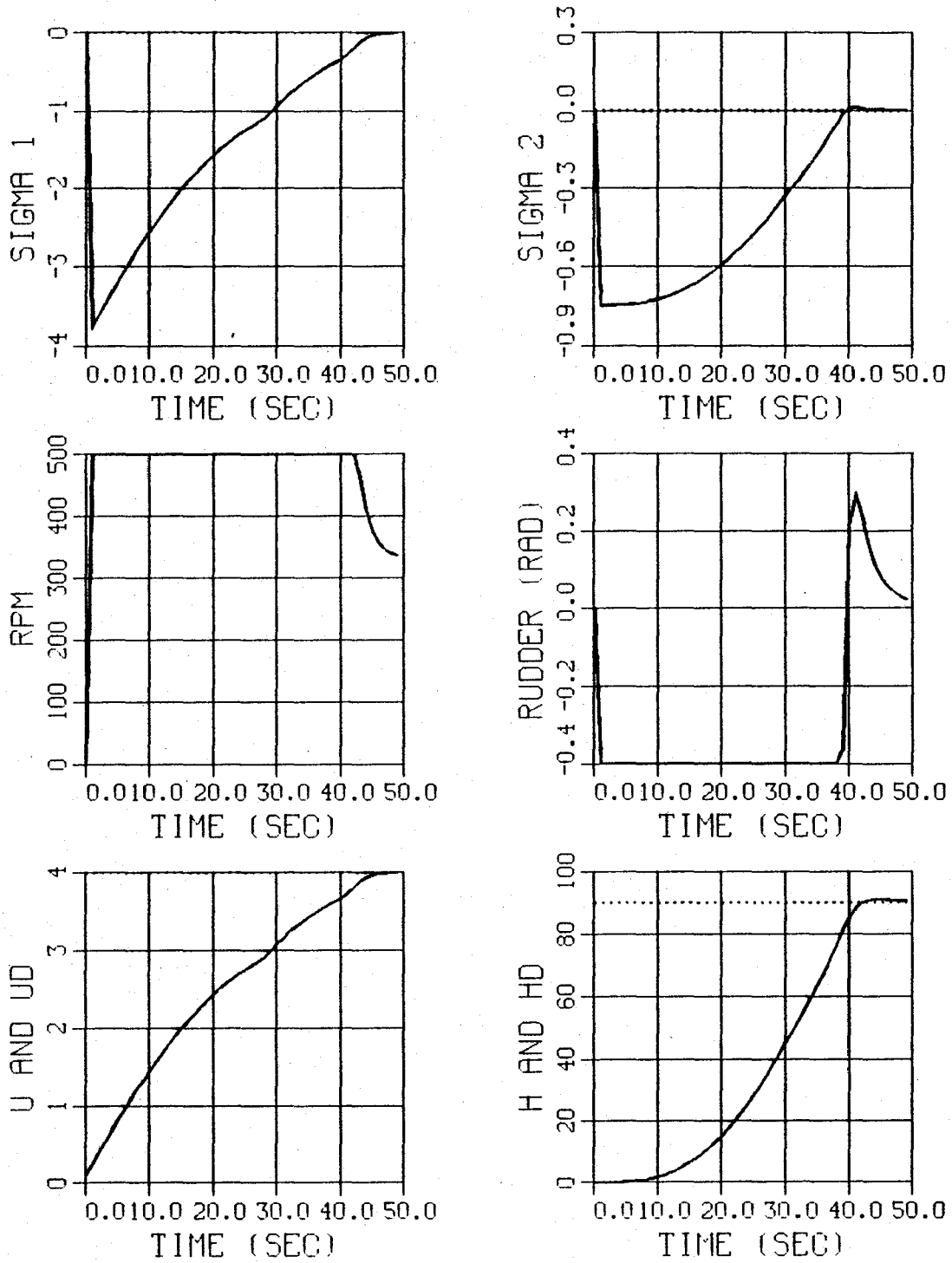


Figure 10. Simulation Including Depth Change (Nonlinear AUV)

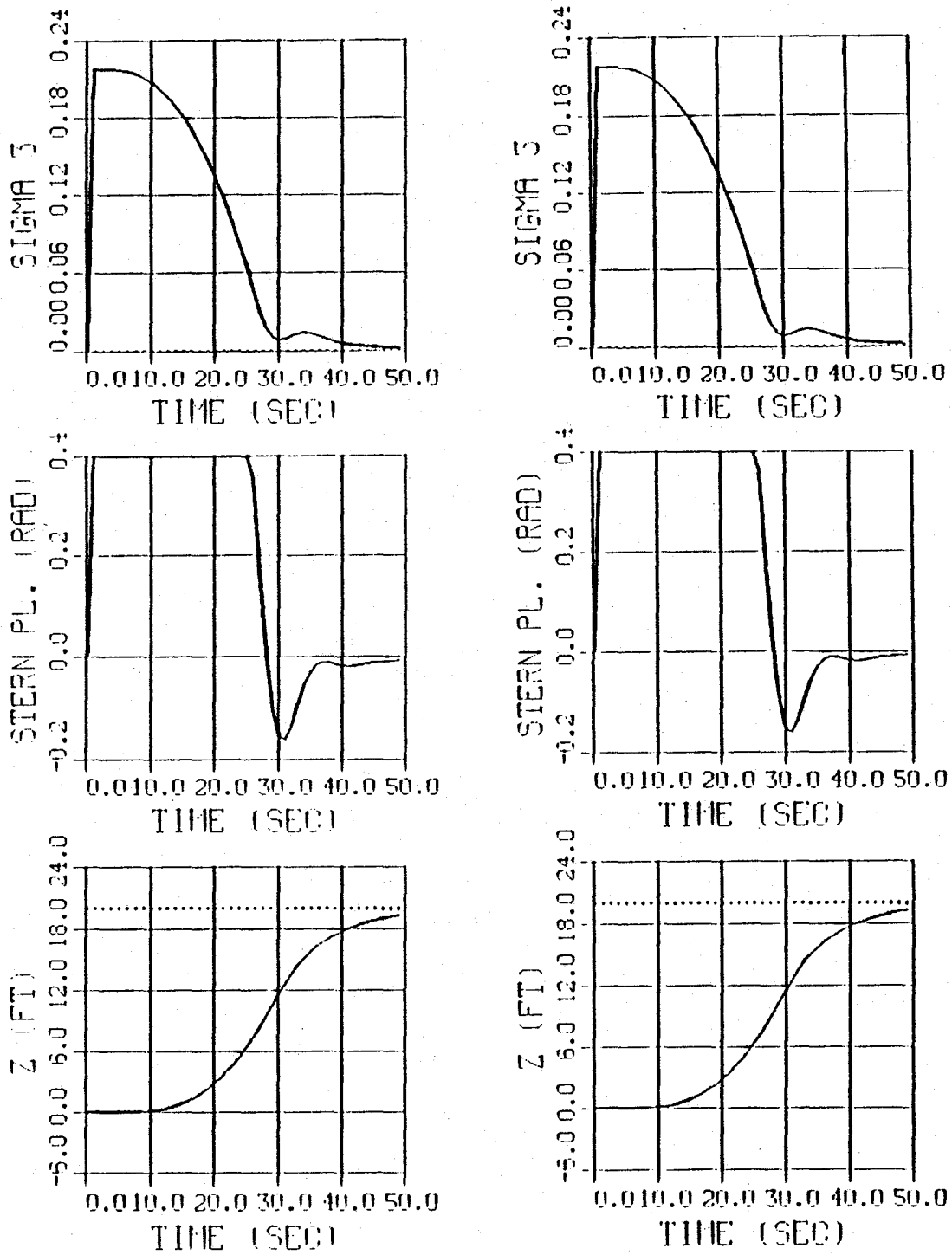


Figure 11. Effect of Tuning on Depth Control

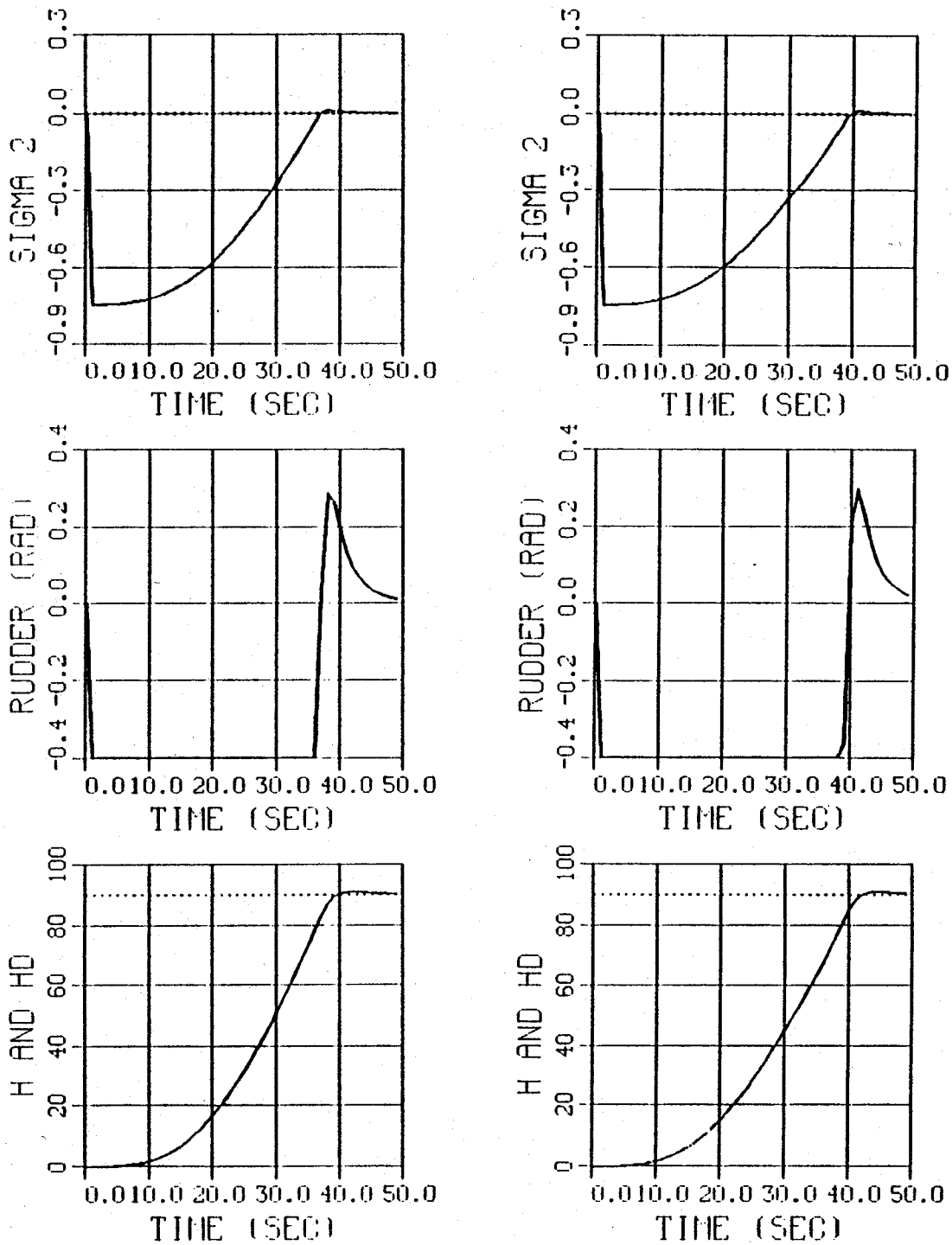


Figure 12. Effect of Depth Changes on Heading Control

that heading control is not adversely affected by a simultaneous change in depth.

E. CONCLUSION

In view of these results it can be concluded that the desired three dimensional control of the full six degree of freedom nonlinear vehicle can be accomplished by 3 SISO systems thus avoiding the need to model the AUV as a MIMO system.

Some problems still remain however that will be addressed in subsequent chapters. First, the control of heading is based on assuming full state feedback which includes sway velocity. To be realistic, since sway velocity is not directly measurable, an observer will have to be designed. In addition, a more global objective is to ultimately cause the AUV to travel to a desired way point rather than simply achieving a desired heading. Though our approach of developing the control laws as separate SISO systems has been validated at this point, it still remains to develop a guidance scheme that can interface with these control laws to achieve path following in the horizontal plane.

The design of the sway velocity observer is dealt with in Chapter 4 and the issue of guidance in Chapter 5.

IV. SWAY VELOCITY OBSERVER DESIGN

A. INTRODUCTION

The heading control law as presented in the previous chapters is based on full state feedback. In reality, the sway velocity (v) will not be measured thus requiring an observer to be designed that can estimate its value. The approach that will be taken is to utilize observer theory to obtain the necessary equations for a reduced order observer and then use the Matrix-x software program to obtain the necessary coefficients for these equations. The depth control law as shown for the runs thus far utilized a full order observer [Ref. 1]. Since the speed control law must be based on full state feedback, sway velocity is the only state variable which requires an observer at this point.

B. OBSERVER DESIGN FOR LINEAR AUV

The state space equations for SISO2 is given in Chapter 2 by Equation 2.9. The state space vector is rearranged to separate the measured from the unmeasured states resulting in equation (4.1).

$$\begin{pmatrix} \dot{r} \\ \dot{\Psi} \\ \dot{v} \end{pmatrix} = \begin{bmatrix} d_{21} & 0 & d_{22} \\ 1 & 0 & 0 \\ d_{11} & 0 & d_{12} \end{bmatrix} \begin{pmatrix} r \\ \Psi \\ v \end{pmatrix} + \begin{pmatrix} d_{23} \\ 0 \\ d_{13} \end{pmatrix} \delta_r \quad (4.1)$$

To develop the reduced order observer, the state space matrices will be partitioned as shown in equation (4.1) and rewritten as follows.

$$\begin{pmatrix} \dot{r} \\ \dot{\psi} \\ \dot{v} \end{pmatrix} = \begin{bmatrix} AO_{11} & AO_{12} \\ AO_{21} & AO_{22} \end{bmatrix} \begin{pmatrix} r \\ \psi \\ v \end{pmatrix} + \begin{pmatrix} BO_1 \\ BO_2 \end{pmatrix} \delta_r \quad (4.2)$$

The output system of equations is then written in the form $Y=[C]X$ as shown below.

$$(Y) = \begin{pmatrix} r \\ \psi \\ v \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} r \\ \psi \\ v \end{pmatrix} \quad (4.3)$$

Partitioning the C matrix and the state vector X_0 to separate the measured and unmeasured states gives:

$$[C_1] = [I] \quad (X_{o1}) = \begin{pmatrix} r \\ \psi \end{pmatrix} \quad (X_{o2}) = (v) \quad (4.4)$$

At this point the relationship shown in reduced order observer theory can be systematically applied to complete the observer design [Ref. 8]. Note, X_{o1} represents the measured state variables and X_{o2} is the sway velocity that is to be observed. Consequently, the following equation is obtained.

$$\hat{X}_{o2} = \hat{v} = [LL] (Y) + [ZZ] \quad (4.5)$$

In order to better understand the following relationships that are used to find $[LL]$ and $[ZZ]$, a block diagram of the observer is shown in Figure 13.

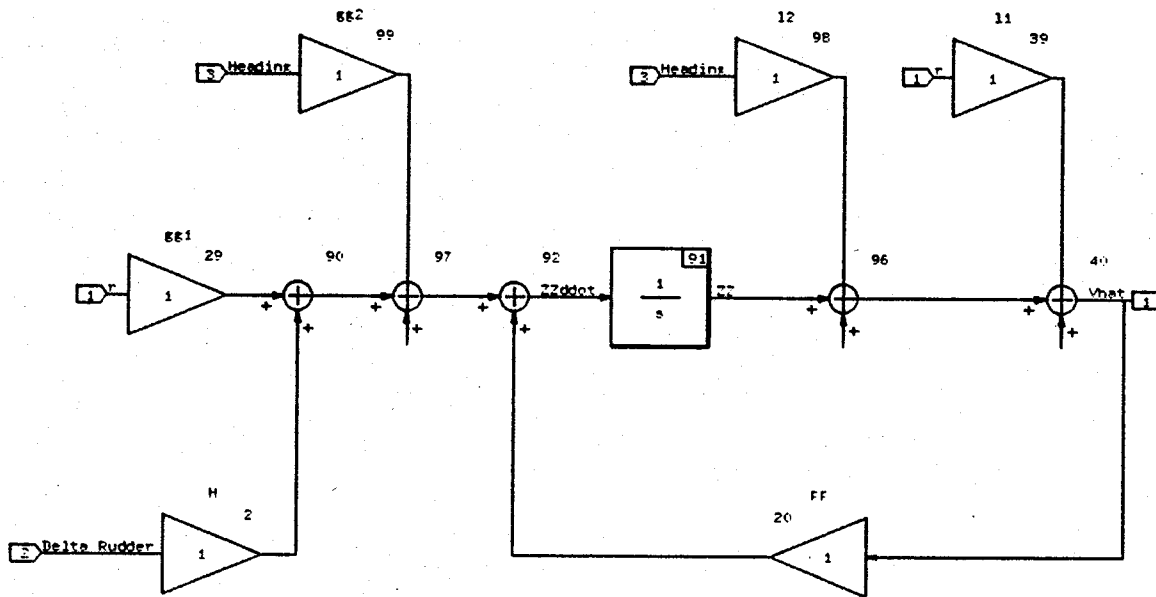


Figure 13. Sway Velocity Observer (Initial Design)

The values of $[FF]$, $[GG]$, and $[H]$ are given by the following equations.

$$[FF] = [AO_{22}] - [LL][C_1][AO_{12}] \quad (4.6)$$

$$[H] = [BO_2] - [LL][C_1][BO_1] \quad (4.7)$$

$$[GG] = ([AO_{21}] - [LL][C_1][AO_{11}]) [C_1]^{-1} \quad (4.8)$$

It will be necessary to consider [LL] and [GG] to have two parts resulting in [LL] = [l₁, l₂] and [GG] = [gg₁, gg₂]. Upon substituting into the above relationships the following equations are obtained.

$$FF = d_{12} - l_1 d_{22} \quad (4.9)$$

$$H = d_{13} - l_1 d_{23} \quad (4.10)$$

$$gg_1 = d_{11} - l_1 d_{21} - l_2 \quad gg_2 = 0 \quad (4.11)$$

Notice that the above values are all scalars and that [LL] is the only part of the observer equation yet to be determined. [LL] will be found using pole placement as follows:

$$|S - FF| = |S - d_{12} - l_1 d_{22}| \quad (4.12)$$

Since l₂ is not included in equation 12 it can be considered arbitrary and assigned a value of zero. The resulting equation for l₁ = [LL] is:

$$l_1 = (-FF + d_{12}) / d_{22} \quad (4.13)$$

It is at this point that Matrix-x was used to determine the coefficients necessary to calculate the observed value of sway velocity. In addition simulations were conducted to compare the observed value of sway velocity to the actual value. The best results were

obtained by choosing FF to have a value of -2 giving the following equation.

$$\dot{\varphi} = I_1 r + ZZ \quad (4.14)$$

Where:

$$ZZ\dot{=} = H \delta_r + gg_1 r + FF \varphi$$

$$FF = -2 \quad gg_1 = -57.827 \quad H = -15.575 \quad I_1 = -104.850$$

The Matrix-x system build block diagram which was used for the simulation is shown in Figure 14 and the entire autopilot with the observer included is shown in Figure 15.

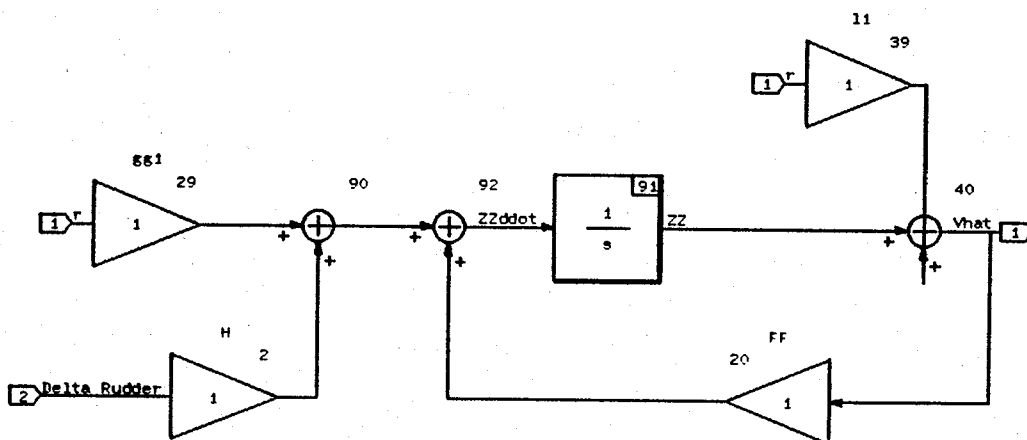


Figure 14. Reduced Order Sway Velocity Observer

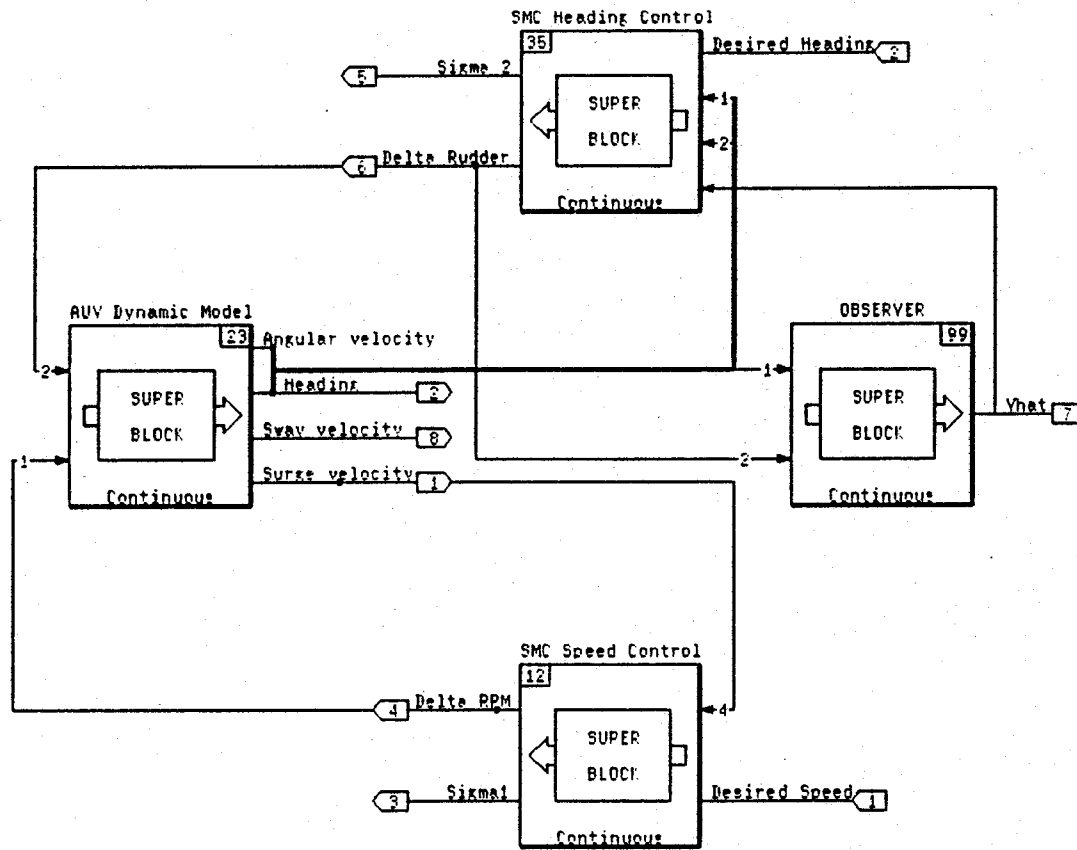


Figure 15. Autopilot With Sway Velocity Observer

The same maneuver simulated in Chapter 2 and shown in Figure 5 was again simulated using the estimated sway velocity. A comparison of sway velocity and its observed value is given by Figure 16. The observed value is exactly the same as the actual showing that the observer has been designed properly. The rest of the run is

included as Figure 17 confirming that the autopilot with the reduced order observer is operating properly. The next step is to include the sway velocity observer into the Fortran code representing the nonlinear AUV.

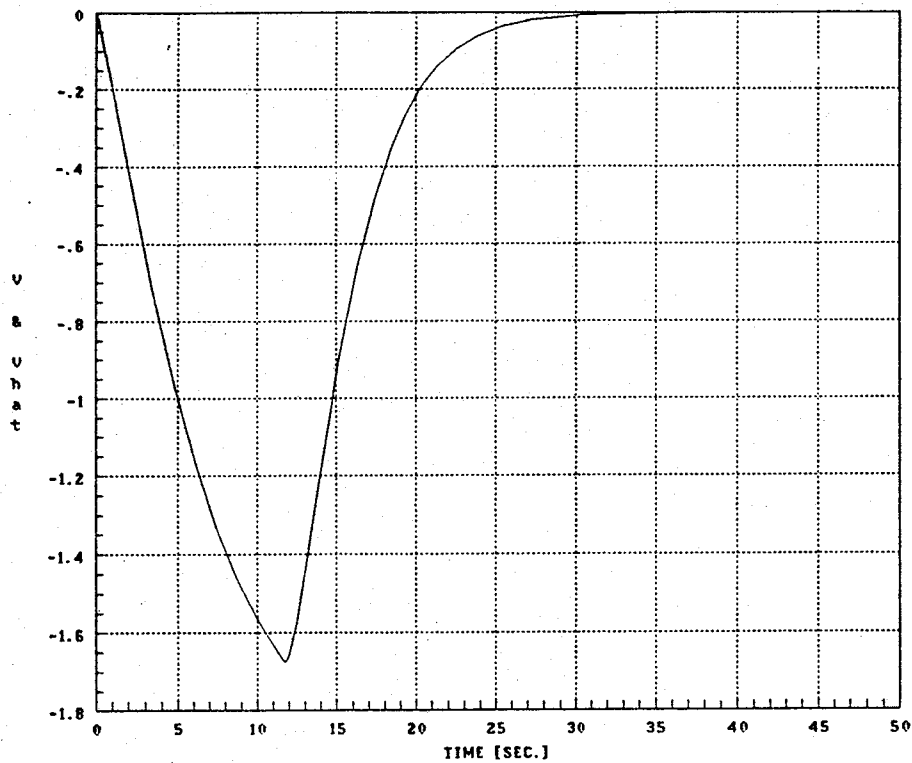


Figure 16. Comparison of Modeled vs Observed Sway Velocity (Linear AUV)

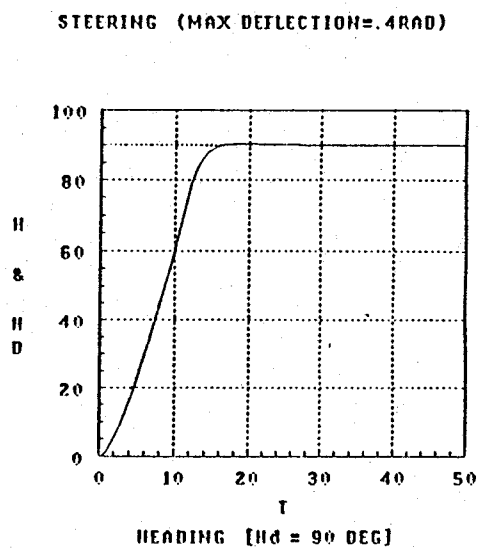
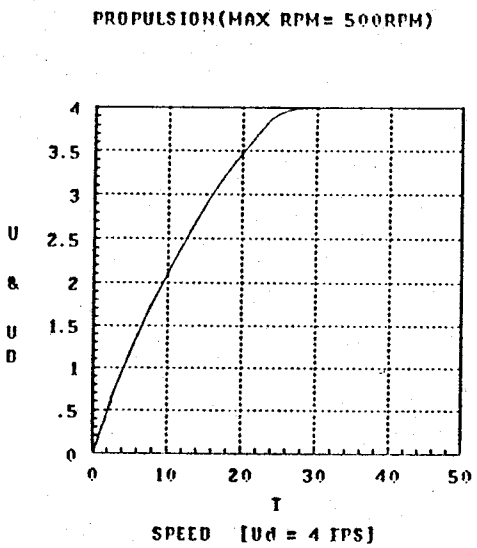
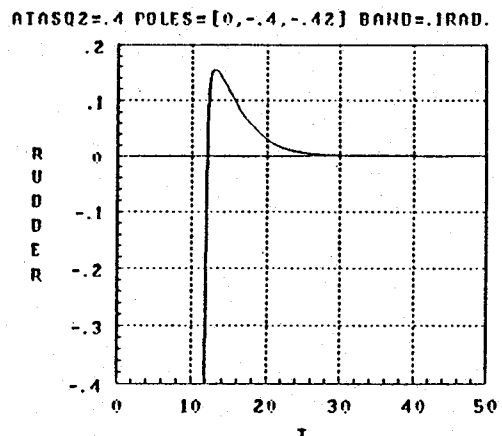
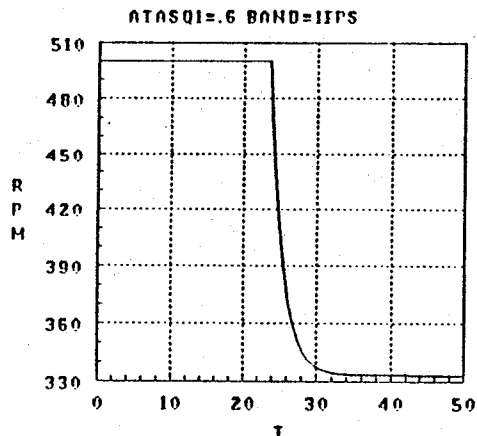
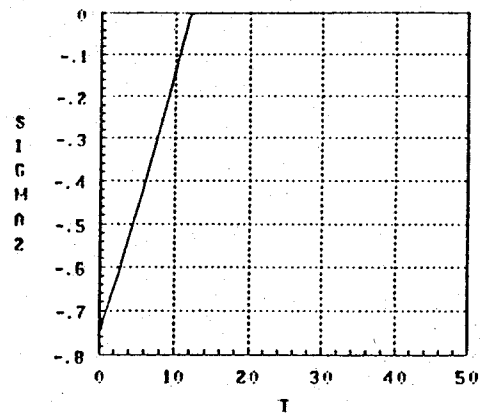
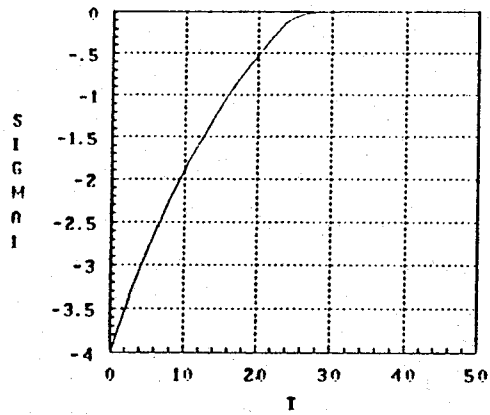


Figure 17. Simulation Using Autopilot With Observer

C. APPLICATION TO NONLINEAR AUV

The initial attempt to use the observer equation was unsuccessful when applied to the nonlinear AUV. The clue to the problem came by noticing that the only time the observer worked well was if the initial value for the surge velocity (u) and the command speed were both 6 ft/sec (the nominal linearization speed). To overcome this problem and allow for a more accurate estimate of the sway velocity throughout its range of operation the Matrix-x model of the AUV was linearized around a variety of speeds and the various coefficients were recorded. Curve fitting techniques were then employed to relate these coefficients to the surge velocity (u) so that the nonlinearities actually present in the real vehicle could be better accounted for. The resulting values for the coefficients are shown in Appendix D along with the equations which resulted. These equations were then included in the Fortran code and the same simulation run was repeated with the improved adaptive observer providing the estimate of sway velocity. The results are shown in Figures 18 and 19. From the simulation it is clear that the control laws continued to work well.

The difference between the observed value and the actual value are shown graphically in Figure 20. This figure shows that the observed value is a fair approximation for v allowing accurate control of the vehicle. The accuracy could possibly have been improved by linearizing around more speeds to get a better set of equations for the coefficients but since the value of v is so small the impact of assuming it to be zero was explored prior to attempting to refine the observer any further.

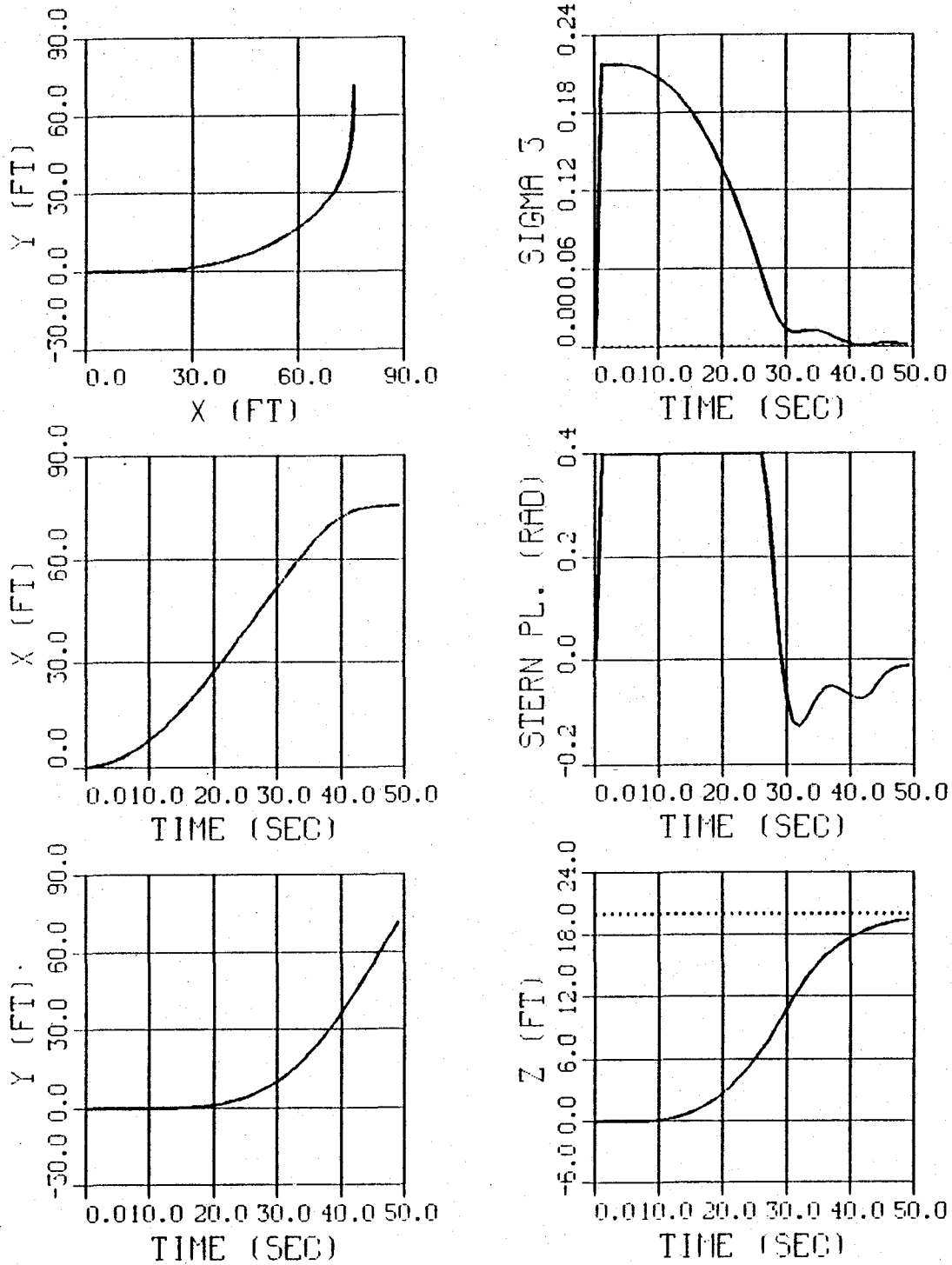


Figure 18. Autopilot With Adaptive Observer (Nonlinear AUV)

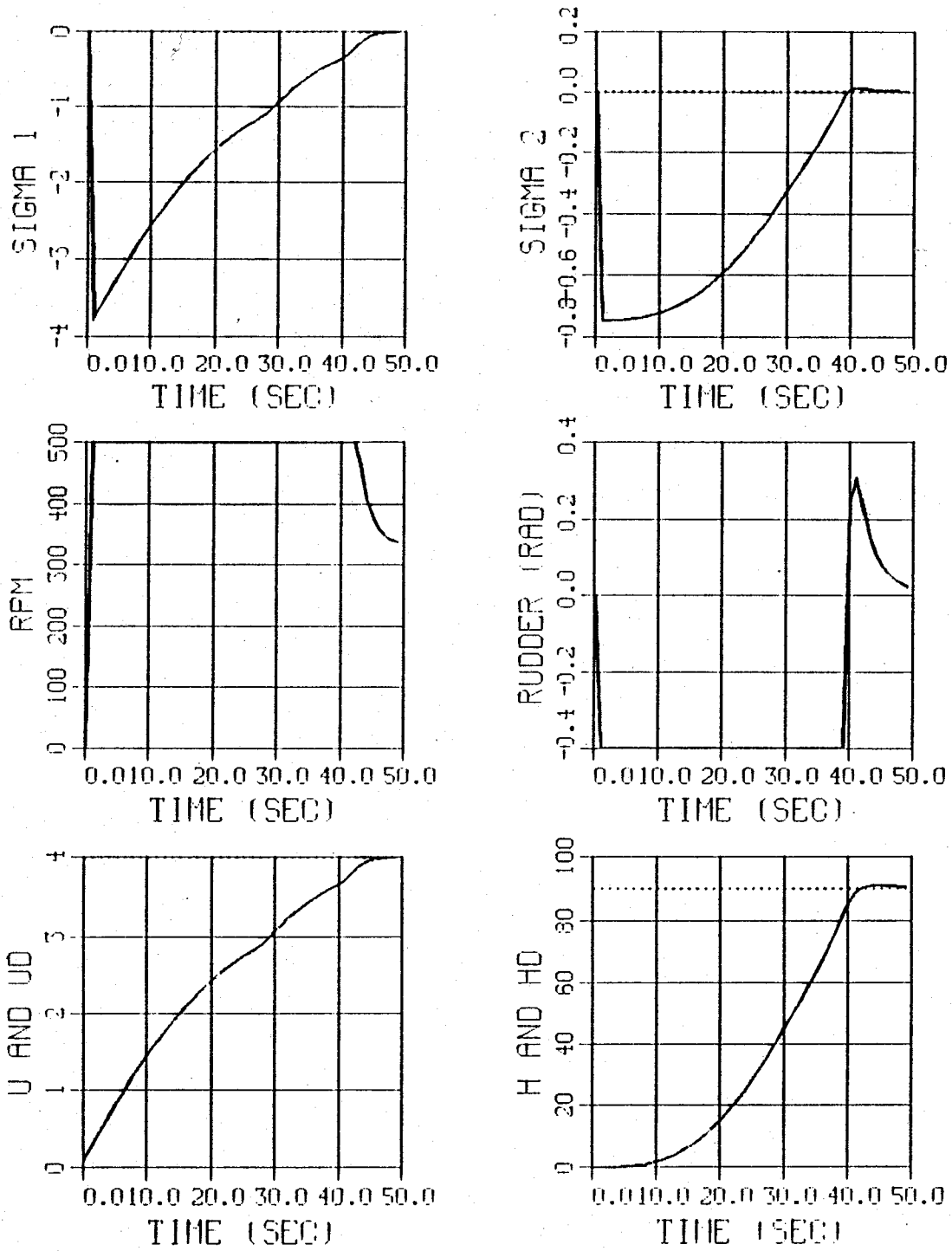


Figure 19. Autopilot With Adaptive Observer (Nonlinear AUV)

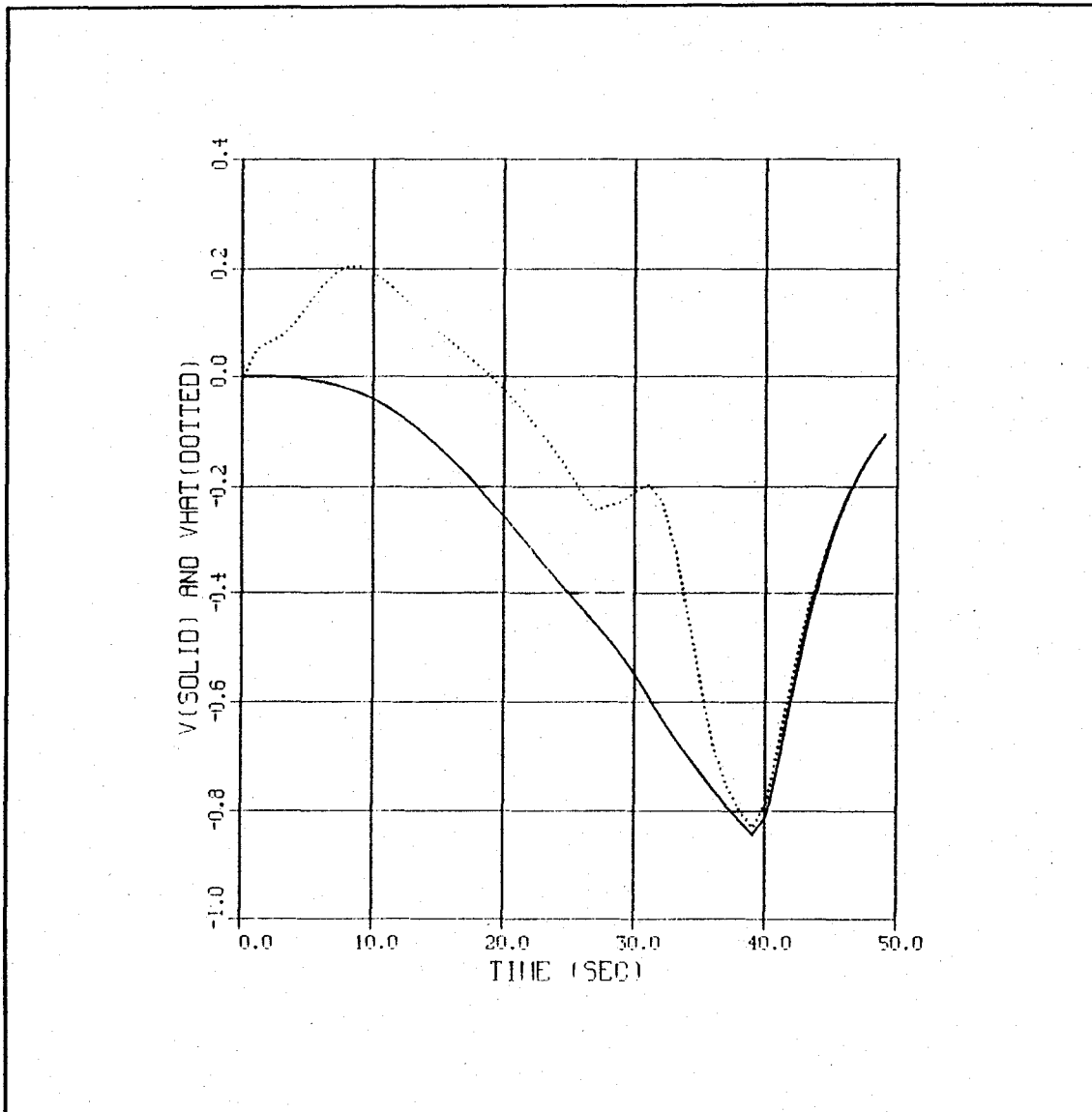


Figure 20. Comparison of Modeled vs Observed Sway Velocity (Nonlinear AUV)

D. EFFECT OF ASSUMING SWAY VELOCITY NEGLIGIBLE

Figures 21 and 22 show the effect of setting the sway velocity equal to zero and commanding the same turn and speed increase as shown in Figure 16. The performance

of the heading control law is virtually the same! It is a tribute to the robustness of sliding mode control that sway velocity is best handled as an unmodeled disturbance than an observed quantity.

E. CONCLUSION

In view of these results, the autopilot will be designed upon the assumption that the sway velocity equals zero. This will simplify the autopilot and increase the speed at which the calculations are performed. Consequently, the problem of improving the observer design is eliminated and SISO2 is effectively reduced to a two state feedback system. This decision may have to be reevaluated if disturbances or currents are added to the model that cause the sway velocity to vary more than can be handled by the nonlinear switching term in the heading control law.

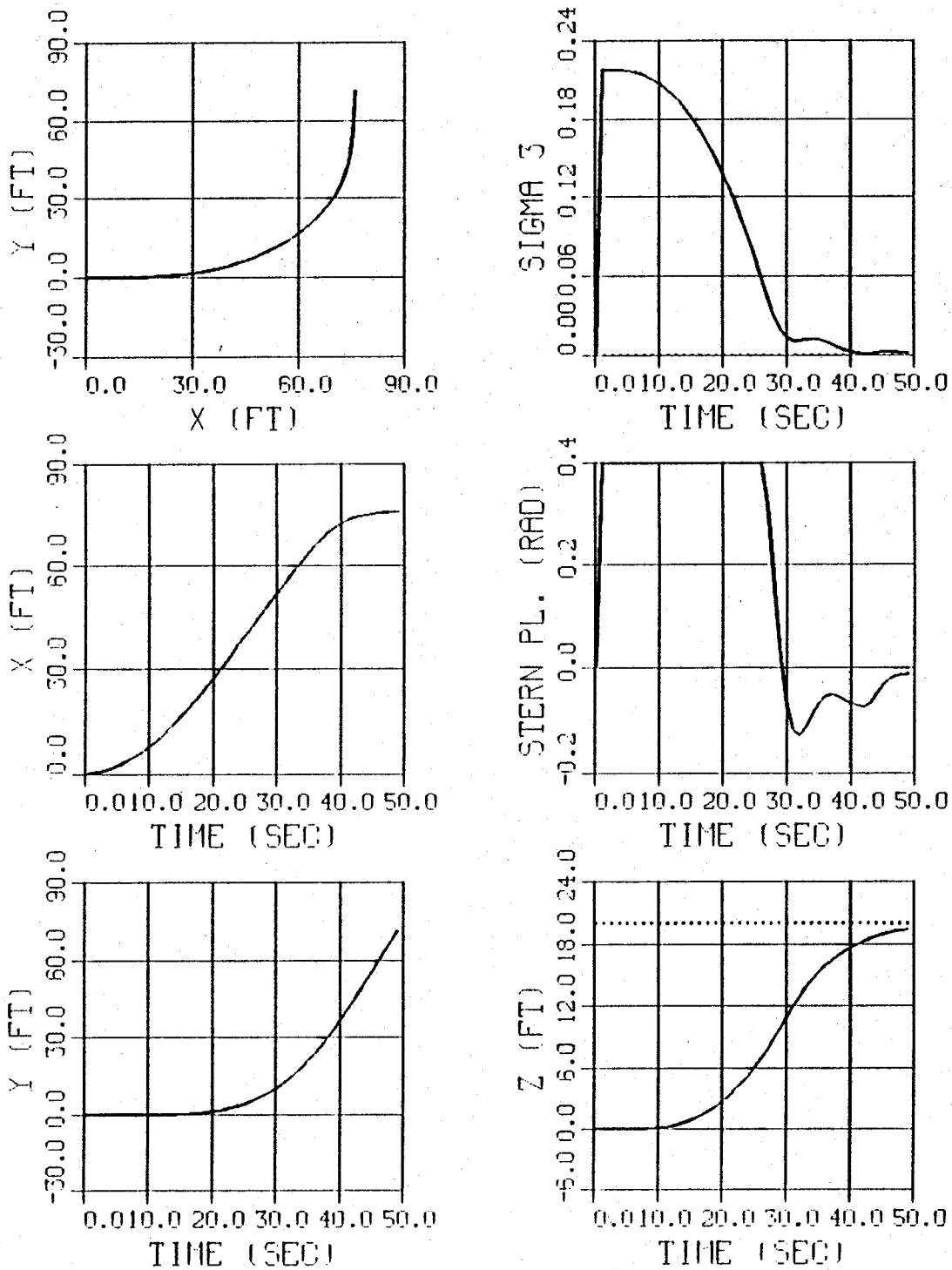


Figure 21. Sway Velocity Set Equal to Zero (Nonlinear AUV)

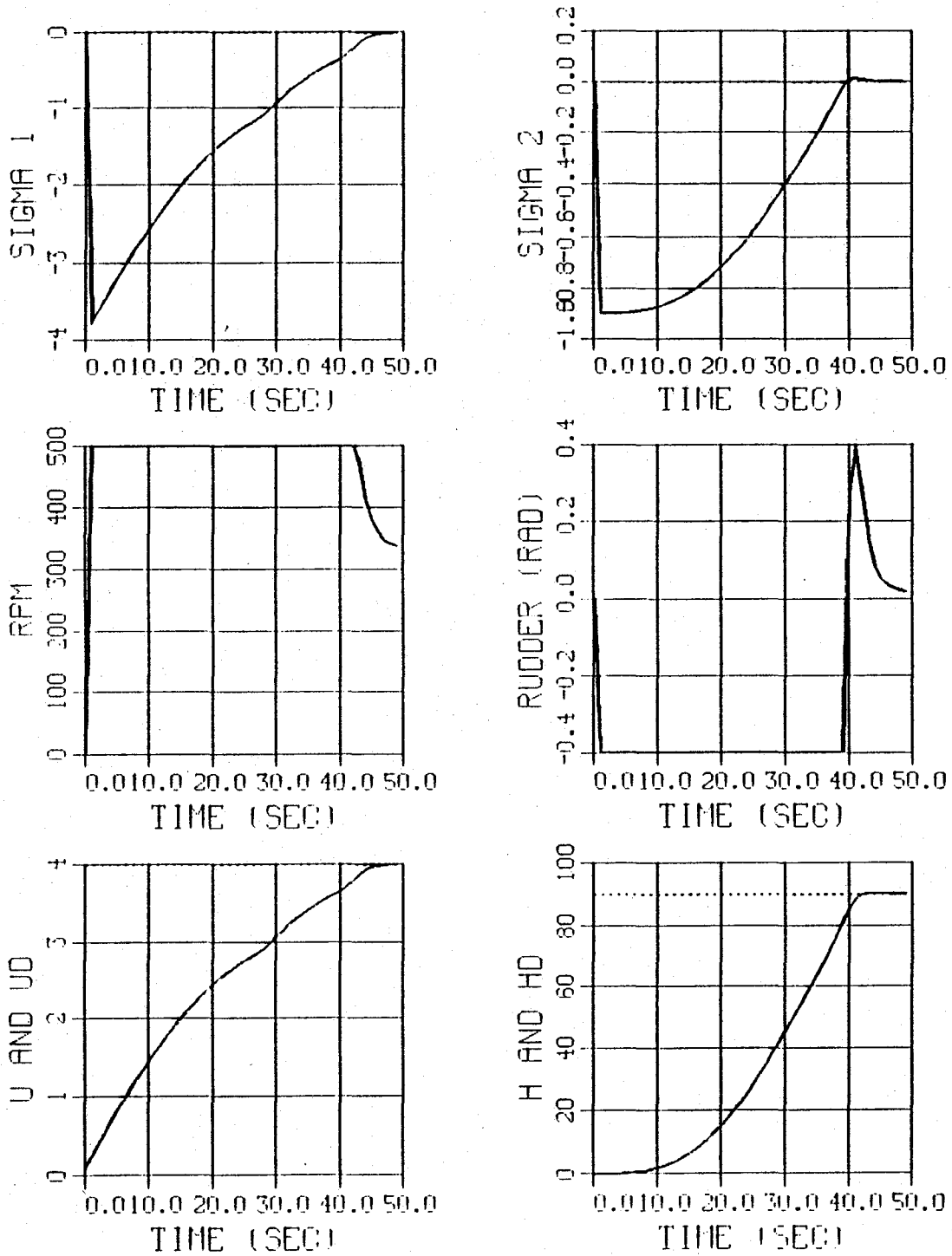


Figure 22. Sway Velocity Set Equal to Zero (Nonlinear AUV)

V. GUIDANCE

A. INTRODUCTION

The guidance portion of an AUV is the link between the mission planner and the autopilot. The mission planner decides the coordinates of the way points the vehicle must travel to and the vehicle's speed. The autopilot, as explained in the previous chapters, requires inputs of desired heading, depth, and speed. Though speed and depth can be given directly to the autopilot, a guidance scheme that will convert the x and y coordinates of the way point into the required heading command must be developed. The method of guidance that was selected is called Line of Sight Guidance (LOS). In this chapter, the basics of LOS are explained and the critical aspects of an autopilot based on LOS are examined.

B. LINE OF SIGHT GUIDANCE

1. Fundamental Relationships

Line of Sight Guidance determines the desired heading by using the vehicles current position and the coordinates of the next way point. As its name implies, LOS essentially points the AUV at the next way point along the line of sight no matter where the vehicle is located. Notice that some sensors, for example an inertial navigation system, must be available to determine the current position of the AUV. Figure 23 shows how the desired heading (ψ_d) can be simply obtained from the coordinates of the current

position (x,y) and the coordinates of the next way point (x_d,y_d) by using the ATAN function.

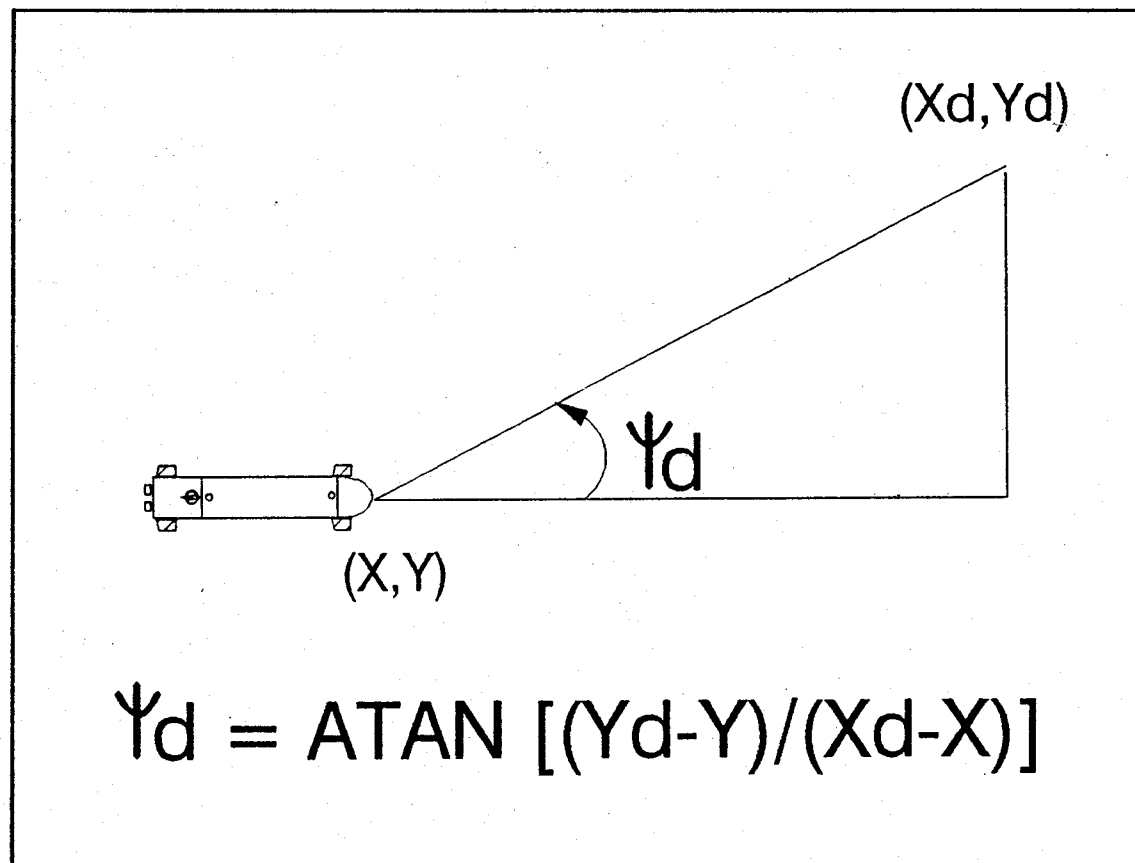


Figure 23. Finding the Desired Heading

Notice that in Figure 23 a turn to port is the obvious choice but if the next way point had been in the fourth quadrant a turn to starboard would have been more efficient. This shows the need to compare which turn will allow the desired heading to be reached with the least amount of course correction. To understand how this was done, it should be pointed out that ψ_d will have a range of values $-\pi < \psi_d < \pi$. To account for the

cases when ψ_d is negative, 2π was added to ψ_d to give ψ_{dp} . Next, a comparison of the absolute value of the angular difference between the desired and the actual heading was made. The choice of ψ_d or ψ_{dp} was based upon which of these gave the smaller course correction. This value that was called ψ_{dm} and was value given to the heading control law as the desired heading. Finally, it should be pointed out, that if the term $(\psi_{dm}-\psi)$ in the heading control law is positive a turn to port will result. Consequently, if it is negative the AUV will turn to starboard. An illustration of the selection of the best turn direction is outlined in Figure 24.

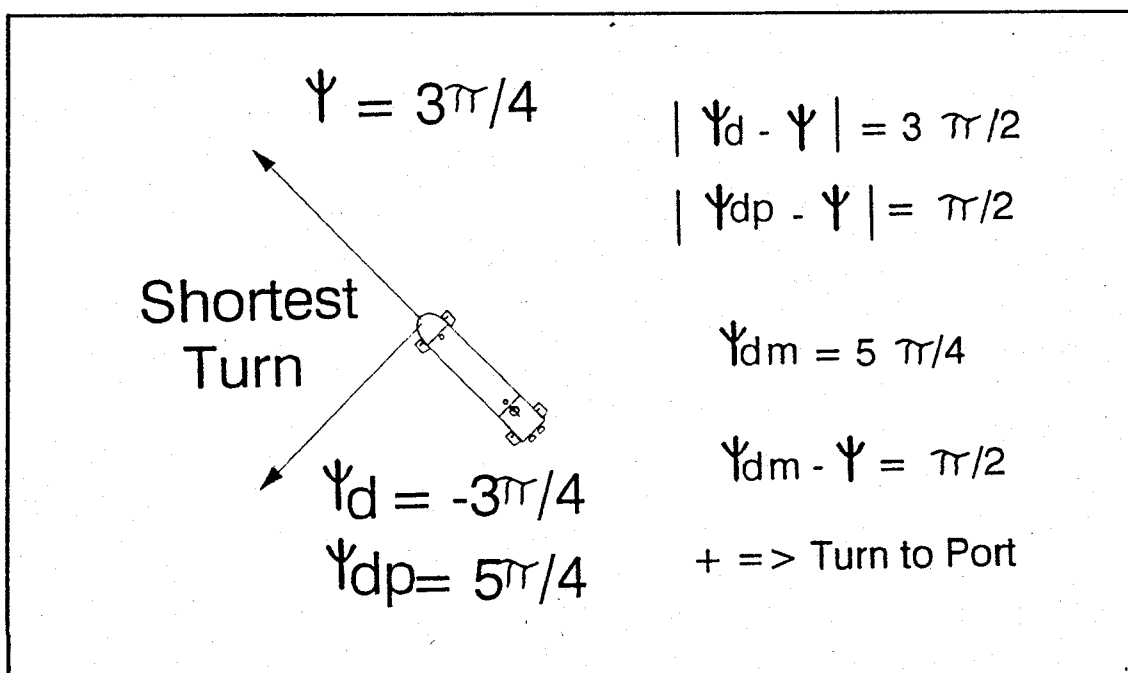


Figure 24. Choosing the Smallest Heading Correction

2. Interface parameters effecting performance

There are several factors that impact the equation that determines the desired heading and consequently the effectiveness of the system as a whole. These factors are

discussed at this point so that the results of the upcoming simulations can better be appreciated.

First, how often does the autopilot receive an update as to its current position from the navigational instrumentation on board. This will effect the accuracy of the calculation of ψ_d as will the accuracy of the navigational instrumentation itself.

Next, a criteria must be established for considering the AUV to be considered close enough to the desired way point that the next way point should be provided by the mission planner. This has the effect of aiming at a target with the desired way point at its center and will account for the advance and transfer of the AUV. The dimensions that are established for this "zone of acceptability" will directly affect the amount by which the AUV overshoots one way point on its way to the next.

The turning and diving characteristics of the AUV must also be considered. A way point must be far enough away that the AUV can physically maneuver to reach it. On the other hand, the more frequently the way points are specified the more control the mission planner can exert on where the AUV is while in transit between two way points.

Finally, the environment the AUV is in must be considered. Factors such as currents can greatly affect the accuracy with which the AUV can be controlled. The number of obstacles that must be avoided and hence the greater need to prescribe a path rather than just a way point will also be an important factor to consider.

Keeping all these factors in mind, it is clear that the mission planner, guidance scheme, and autopilot must ultimately be evaluated as a system with a particular mission

rather than three independent functions. It is in this context that the following results are presented.

3. Results

Run 1 shows the versatility of the system using LOS guidance under the following nearly ideal conditions.

- (x,y,z) position updates are accurate and are provided every .1 sec.
- Way points are distant enough to be well within the turning and diving capabilities of the vehicle.
- The target radius is one shiplength.
- No disturbances or obstacles.

The mission that is envisioned is that of a straight line transit followed by a slow speed search. It will require both port and starboard turns and increases and decreases in both depth and speed. The way points are identified in the horizontal plane by crosshairs and the desired depth and speed are represented by dotted lines. Figure 25 shows the results of the simulation. The AUV went to each way point that was specified and did so at the required speed. The control action of the rudder, dive planes, and motor were very efficient. This run has to be considered to be highly successful thus proving LOS to be a valid guidance method. Notice that the mission ends with the AUV one shiplength away from the last way point waiting for the new way point to be provided by the planner.

The first parameter that was changed was the time between updates of the

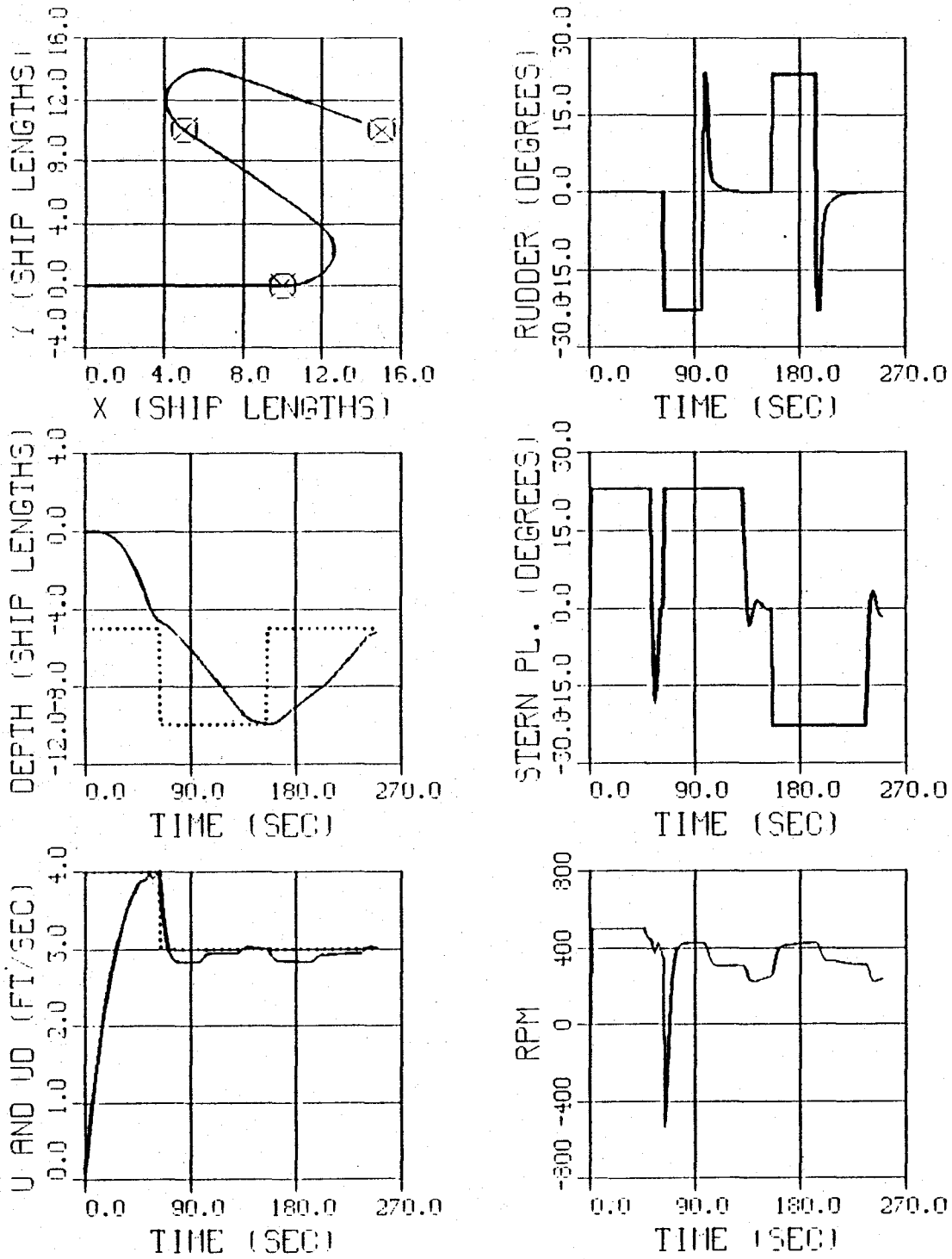


Figure 25. RUN 1, LOS Guidance for the Nonlinear AUV

actual position of the AUV. The first signs of a decline in performance became evident when the update time reached 5 seconds. The results are shown in Figure 26. The control was still very successful which shows how tolerant LOS guidance is to inaccuracies in determining the AUV's position. Notice however that there is an increase in the control effort for the rudder. The fluctuations that result whenever the update is received are not optimal so it is recommended that the update time be kept below 5 seconds.

The update time and the accuracy with which the AUV's position can be determined will become more and more important as the way points become more closely spaced in an effort to achieve even greater control of the transit path. Notice that these way points are approximately ten shiplengths apart.

The next simulation runs explored the effect of the size of the target radius. The optimal target radius was determined to be two shiplengths. The improvement that a two shiplength target radius resulted in as compared to a target radius of one shiplength can be seen by comparing Figures 25 and 27. Since the next way point is provided one shiplength sooner, the control action required to reach it can be initiated earlier. The old way point is still reached by virtue of the momentum of the AUV but the amount by which the AUV overshoots is decreased. This reduces the total path length and consequently improves the overall performance. A target radius of two shiplengths seems to work best since increasing it any further would result in missing the original way point by too great a margin as the AUV redirects itself toward the newly assigned way point.

The effect of the frequency with which the way points are given was explored by the next simulation run. In an attempt to improve the performance of the AUV two

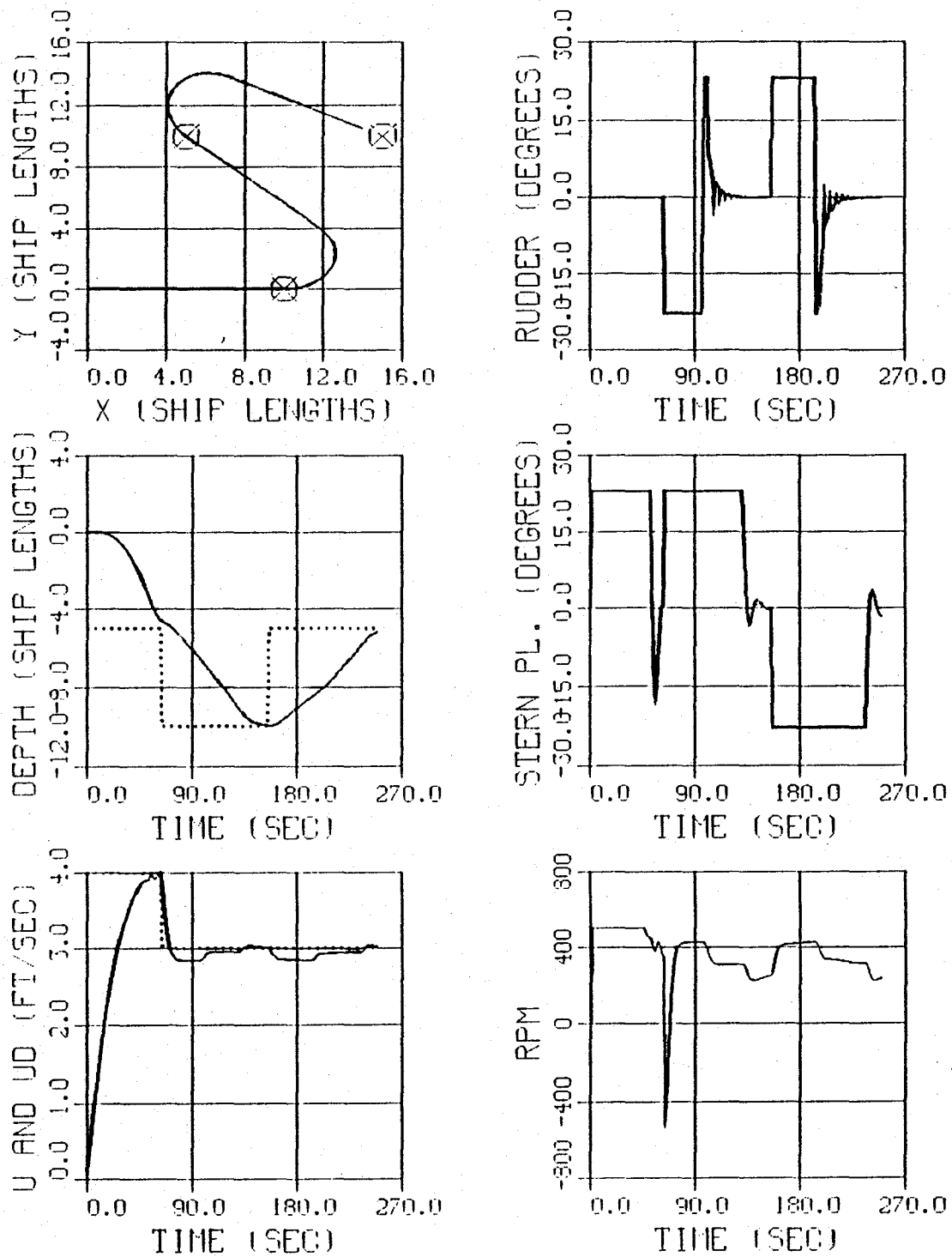


Figure 26. RUN 2, LOS Guidance (Update Every 50 Samples)

additional way points were included in between those originally provided. This was done in an attempt to cause the AUV to take more of a straight line path to the next way point. The additional way points were in line with the original two and were located about five shiplengths away. Five shiplengths is the turning diameter for this vehicle so the idea was to get a maximum turning effort initially and then follow a straight line for the second part of the path. The target radius for this run remained two ship lengths but the position update was 1 second since the way points are closer together. Figure 28 shows the results. A greater degree of control of the AUV is attained and a shorter distance of travel is required. The improvement can be easily seen by comparing Figures 27 and 28.

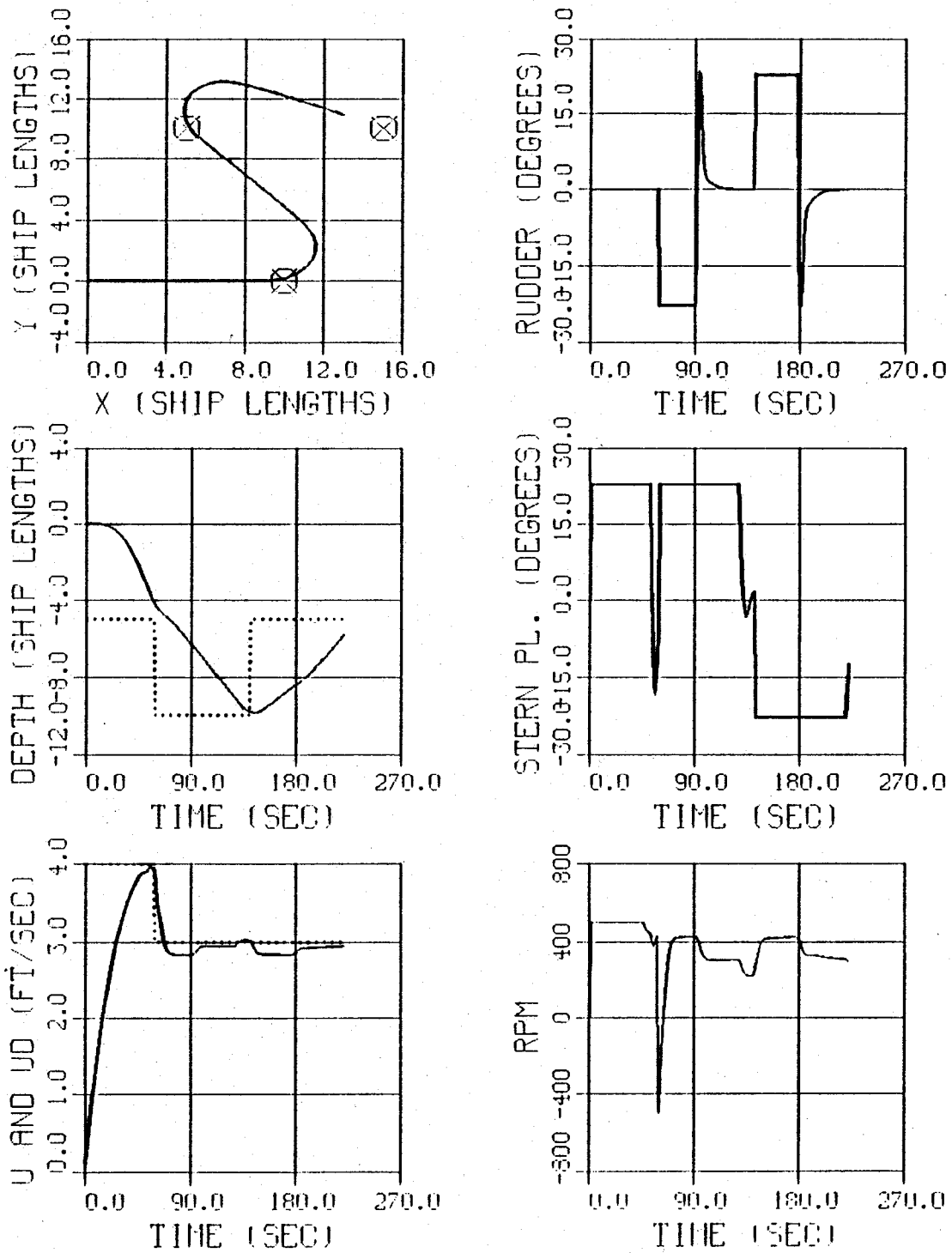


Figure 27. RUN 3, LOS Guidance (Target Radius = 2 SL)

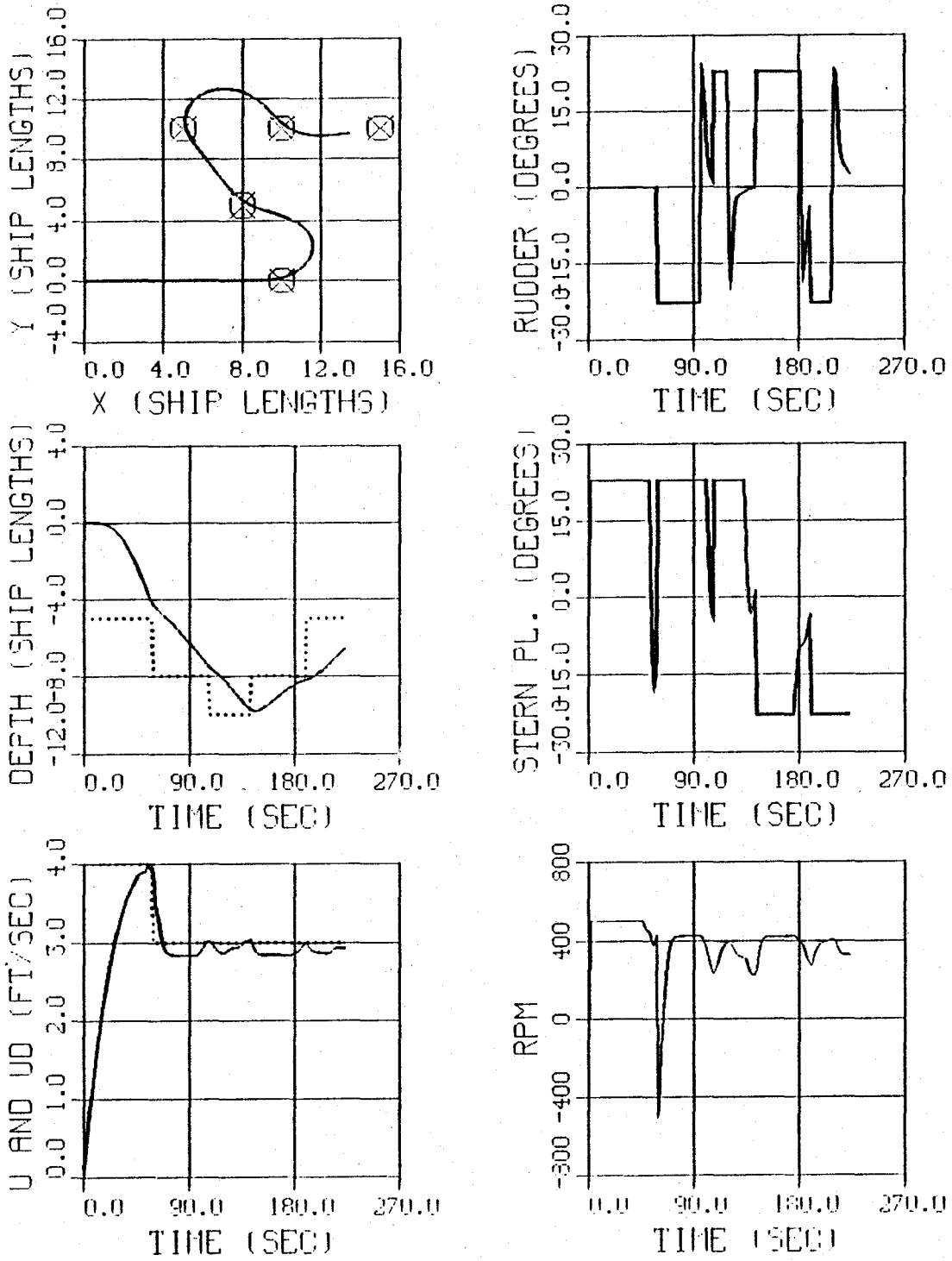


Figure 28. RUN 4, LOS Guidance (Additional Way Points)

VI. HIGH SPEED APPLICATIONS

A. INTRODUCTION

The design of the autopilot and LOS Guidance system is based on a nominal operating speed of 6 ft/sec. This chapter explores the capability of the autopilot and guidance system to successfully control the AUV during high speed operations. The goal is to control the AUV at speeds as high as 50 ft/sec (30 knots) and as low as 3 ft/sec. This wide range of operating speeds will result in an extremely large change in the vehicle's hydrodynamic coefficients. Recalling that the control is based on a model that was linearized around a speed of 6 ft/sec, it becomes clear that this will be a severe test of the robustness of the sliding mode control laws.

To better appreciate the extent of the unmodeled behavior, graphs of pitch and pitch rate are included in the simulation results. Recall that they were both considered to be zero during the simplification of the equations of motion for SISO2.

As was mentioned previously, the depth control law has been based on observed values of pitch and pitch rate up to this point. This is advantageous for a small vehicle that does not have enough room for the instrumentation necessary to measure these values. For an AUV capable of a 30 knot operating speed however, it is assumed that there will be instrumentation onboard to provide these measurements. Consequently, the simulation will be done assuming full state feedback with the exception of the sway

velocity. Sway velocity will still be given a value of zero and handled as a disturbance as was explained in Chapter 4.

B. AUTOPILOT SIMULATION RESULTS

The first set of simulation runs test the ability of the autopilot to achieve the desired heading, speed, and depth. The capability of the LOS guidance system will be demonstrated later once the autopilot has been verified.

1. Run 1

The conditions for Run 1 are as follows:

- Command speed 18 FPS.
- Command depth 2 Shiplengths
- Command heading 90 Degrees
- Integration time step .01 Sec.

This run tested the ability of the autopilot to operate at a speed three times greater than it was designed for. Before satisfactory results could be attained at higher speeds, the integration time step had to be decreased. The relationship between the surge velocity and the integration time step is found by expressing it in a nondimensional time as shown in Equation 6.1.

$$ND\Delta t = u \Delta t / L \quad (6.1)$$

This relationship shows that if the nondimensional time step is to remain constant, the time step used within the simulation must decrease as the speed increases to obtain the correct results. By referring back to the initial work at slow speeds, the value for the nondimensional time step was found to be .1. Based upon Equation 6.1, an integration time step of .01 sec was chosen for the surge velocity of 18 ft/sec.

The results as shown in Figures 29 and 30 are outstanding. The autopilot operates even better at high speed. All desired values are attained with a minimal amount of control effort and with no modification of the control laws. The autopilot is now ready to be verified using the boundaries of its design envelope.

2. Run 2

The integration time step was decreased to .005 seconds for this run at 50 ft/sec to be consistent with keeping the nondimensional time step constant. In addition the Auv is given an initial surge velocity of 50 ft/sec rather than starting from rest to show the effect of a high speed turn. The remaining characteristics of this run are the same as Run 1. The results are shown in Figures 31 and 32.

Again the results are very good with no problems evident. This result is a most impressive verification of the robustness of sliding mode control.

3. Run 3

Run 3 is included to show that the autopilot is also capable of slow speed operations at 3 ft/sec. An integration time step of .1 seconds is adequate for this speed. Figures 33 and 34 show that the AUV is more sluggish at the slower speed but the control laws have no problem attaining the commanded heading, speed, and depth.

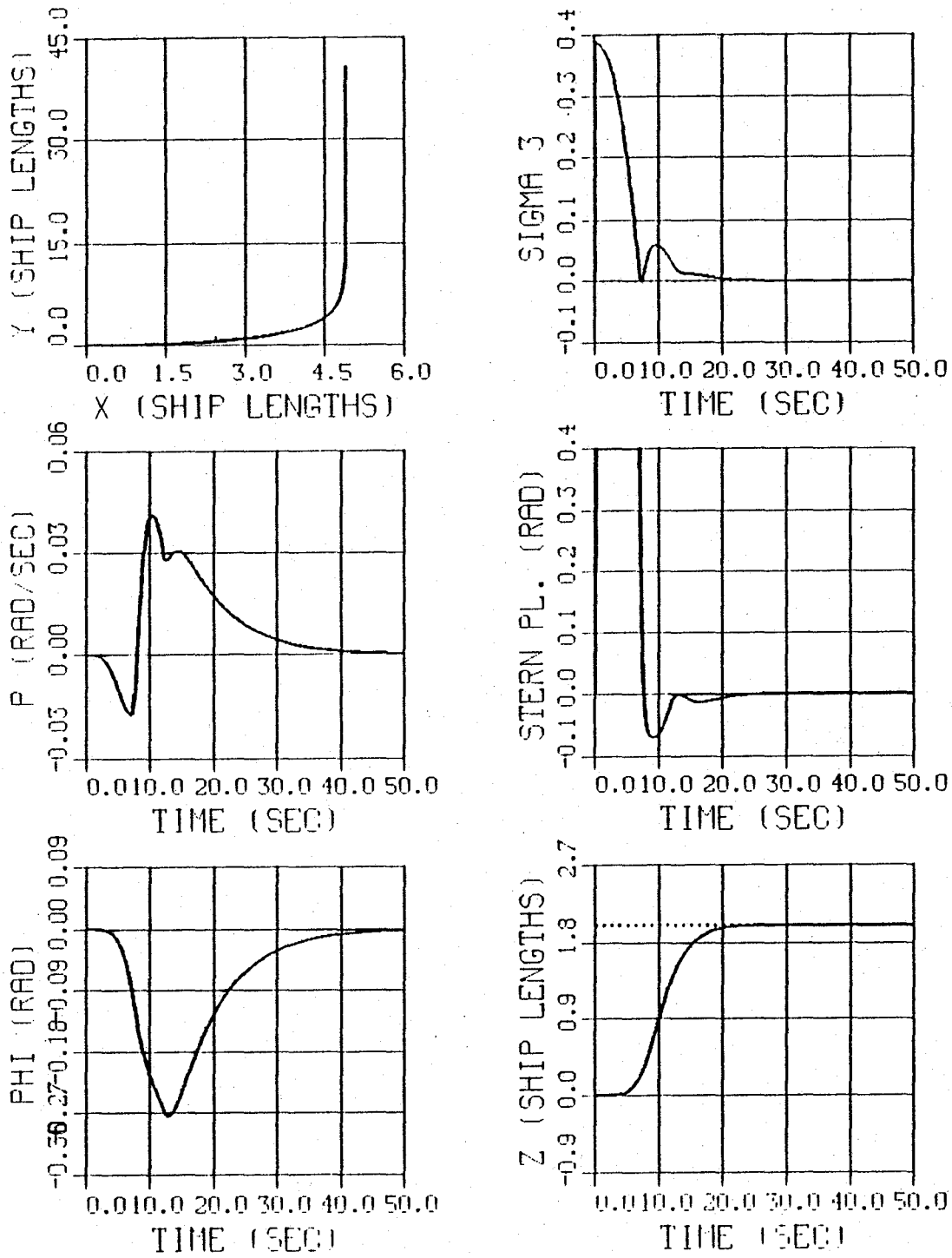


Figure 29. Run 1, High Speed Maneuver (18 ft/sec)

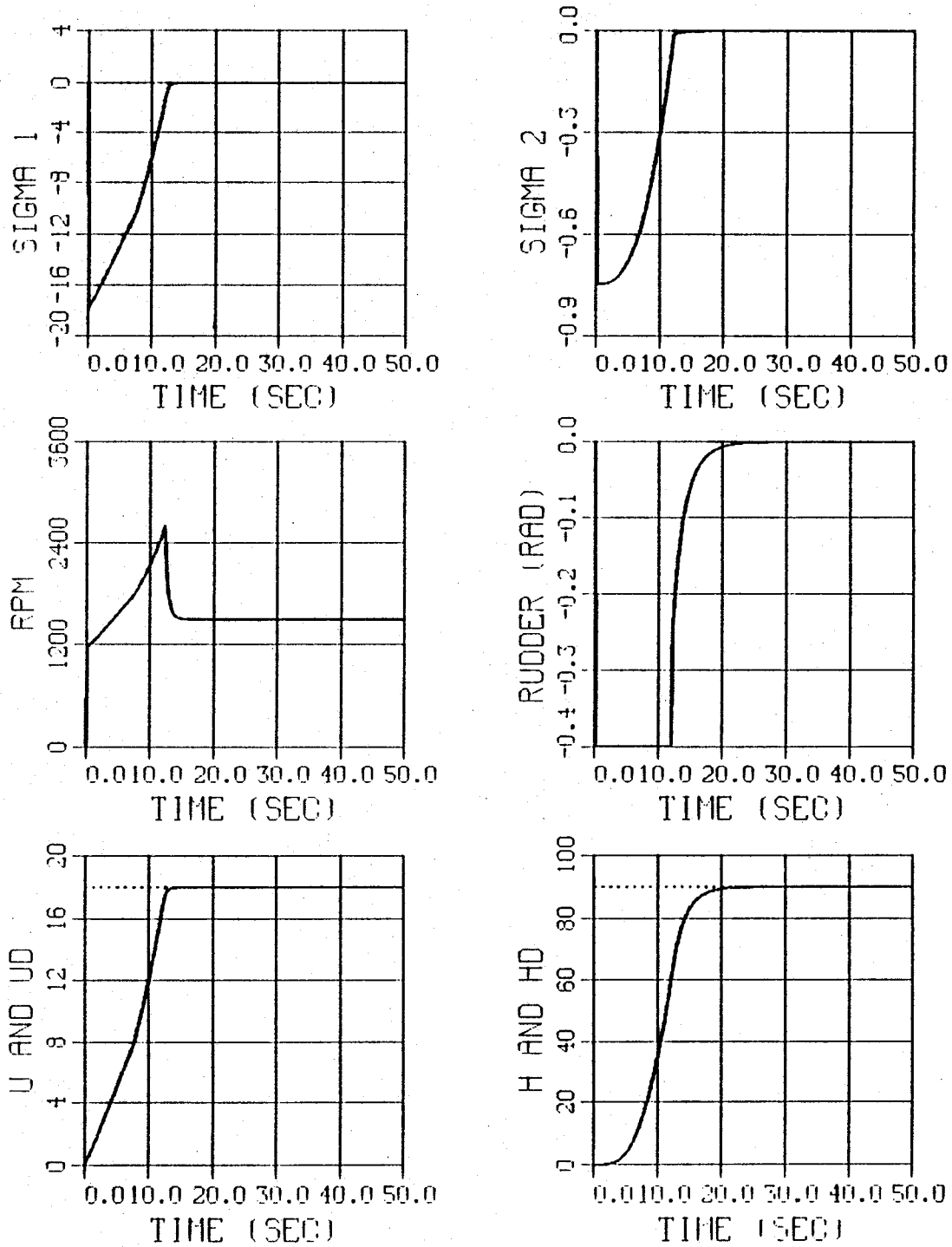


Figure 30. Run 1, High Speed Maneuver (18 ft/sec)

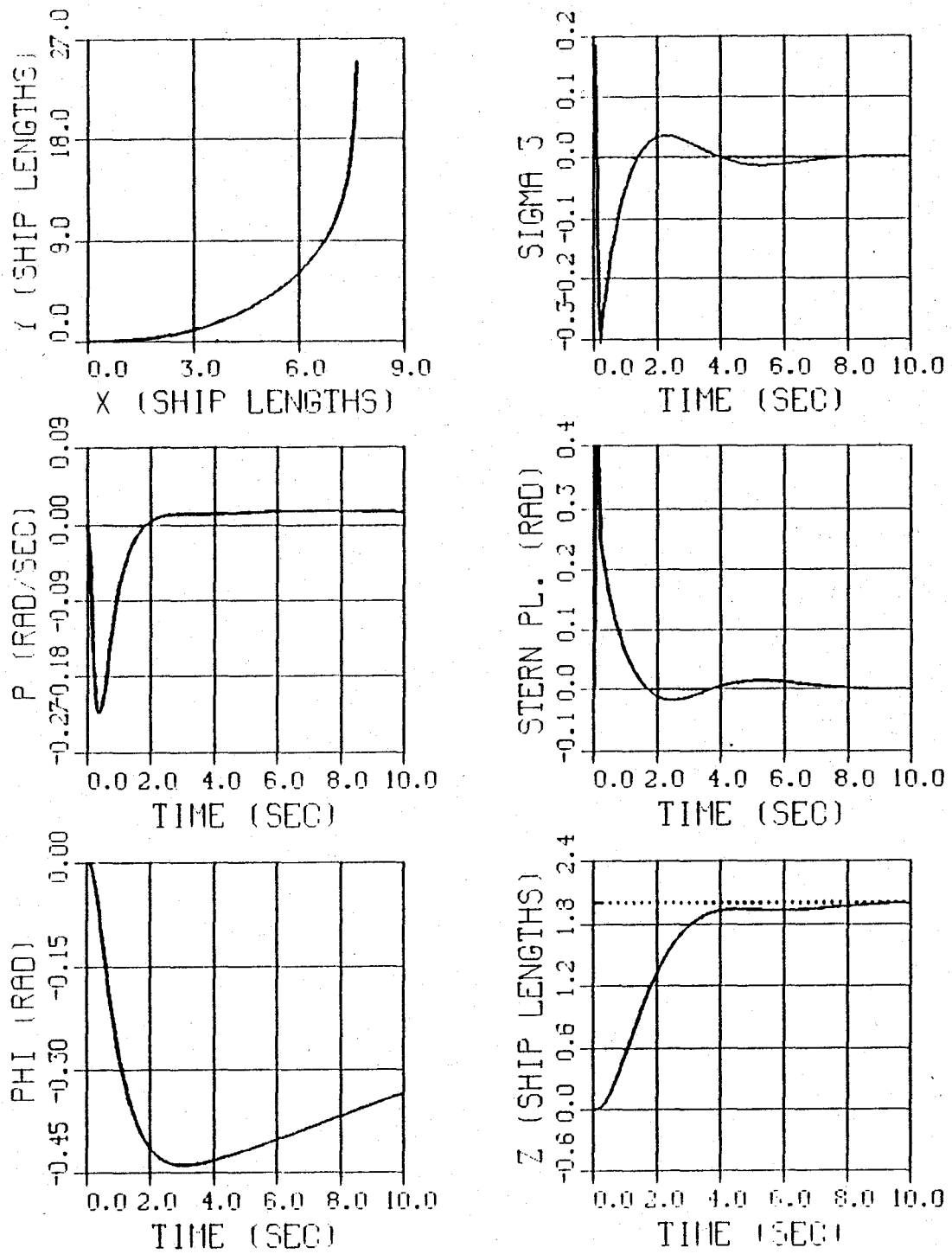


Figure 31. Run 2, Maximum Speed Maneuver (50 ft/sec)

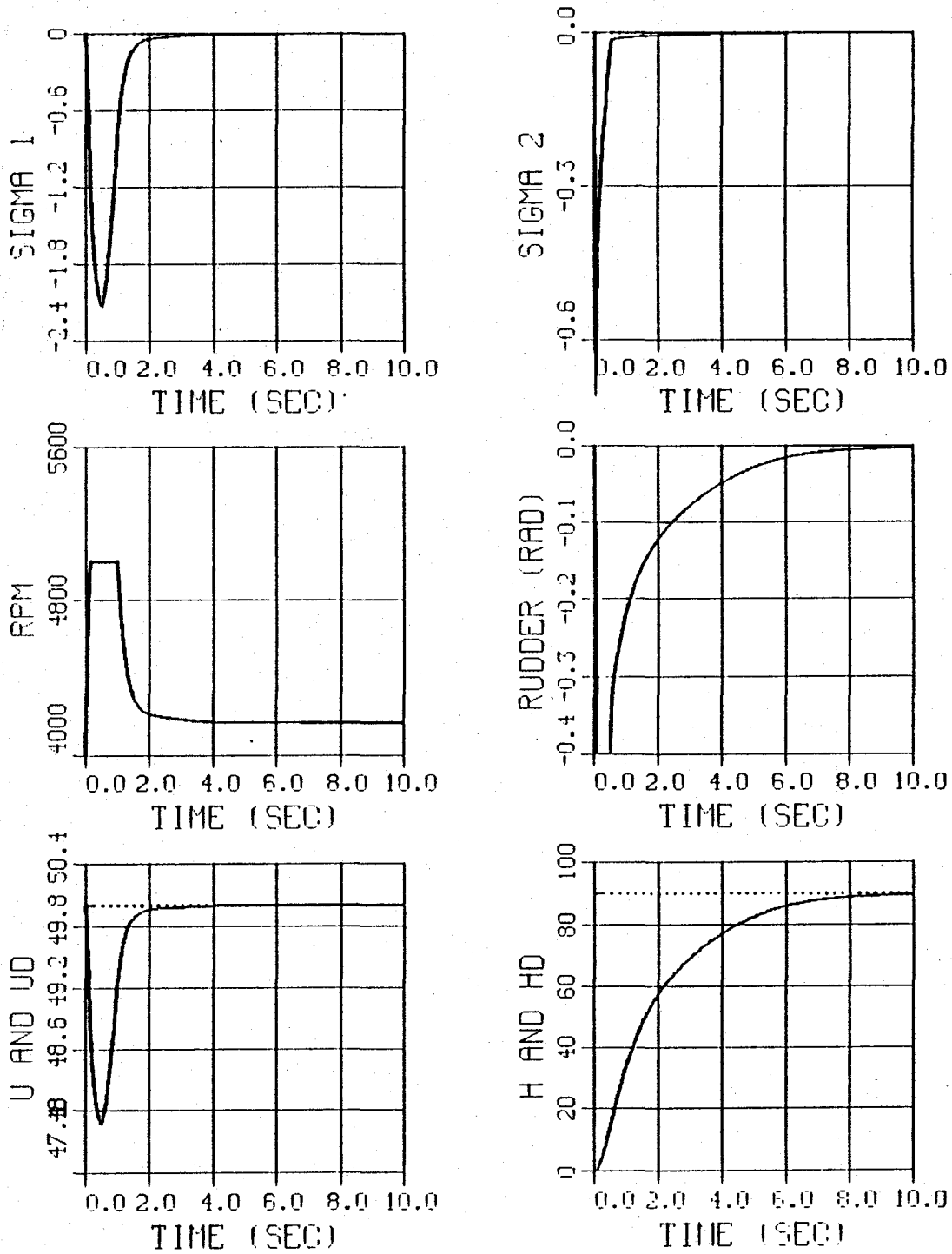


Figure 32. Run 2, Maximum Speed Maneuver (50 ft/sec)

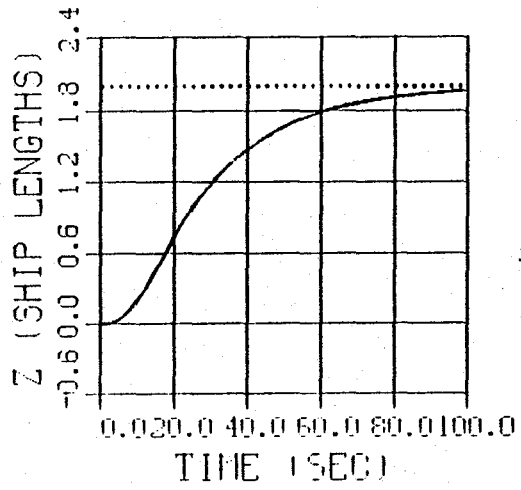
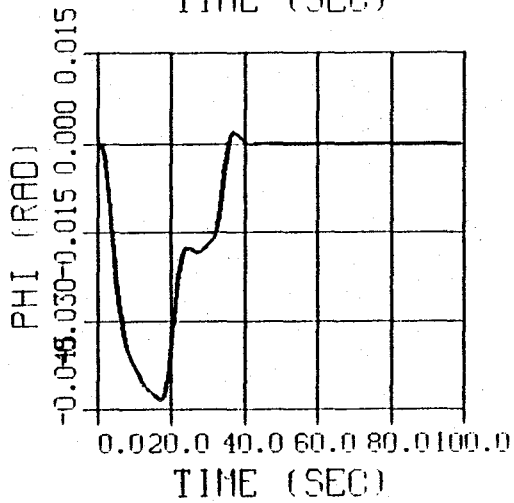
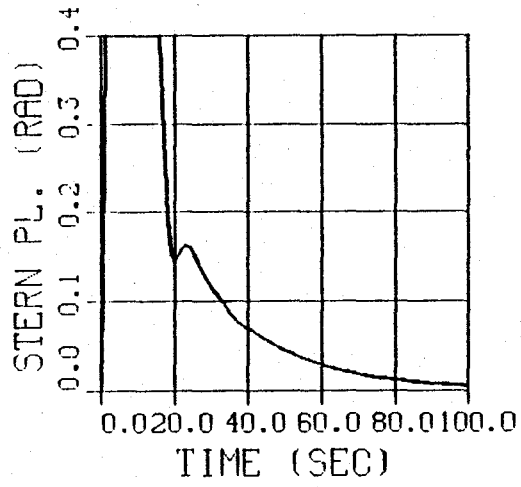
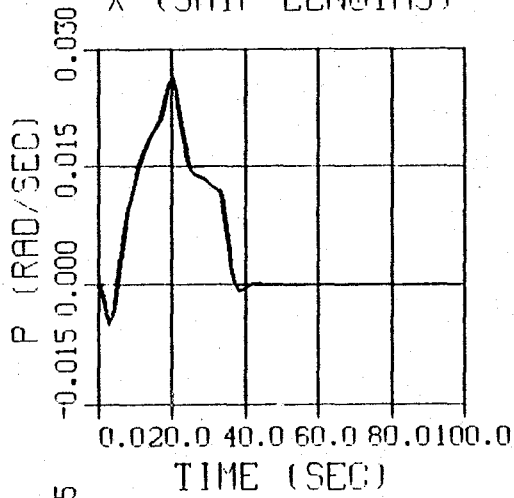
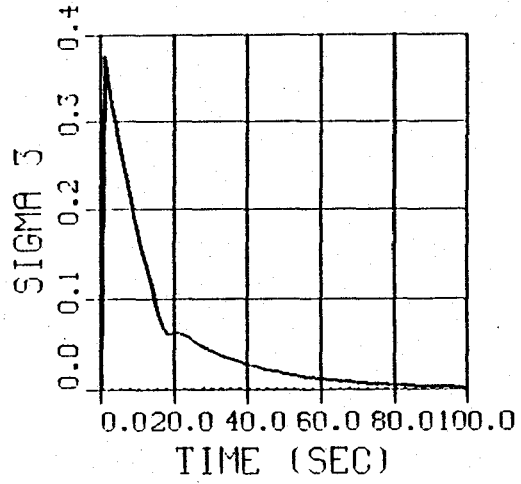
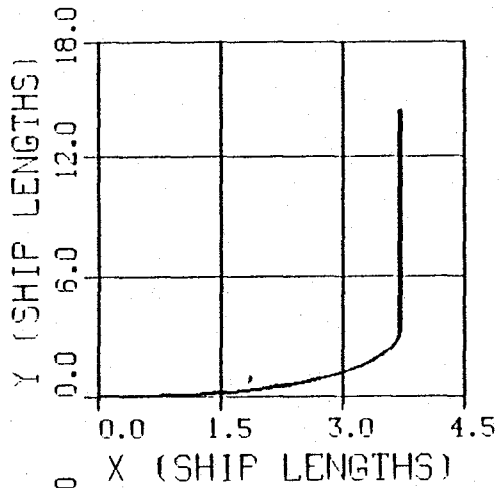


Figure 33. Run 3, Slow Speed Maneuver (3 ft/sec)

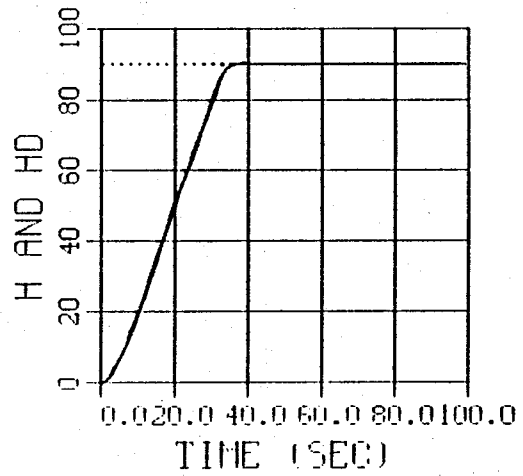
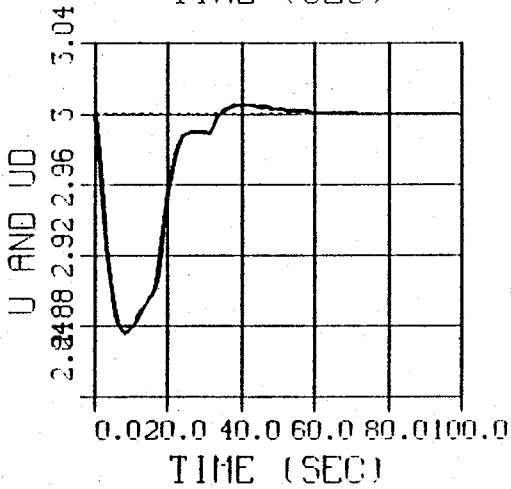
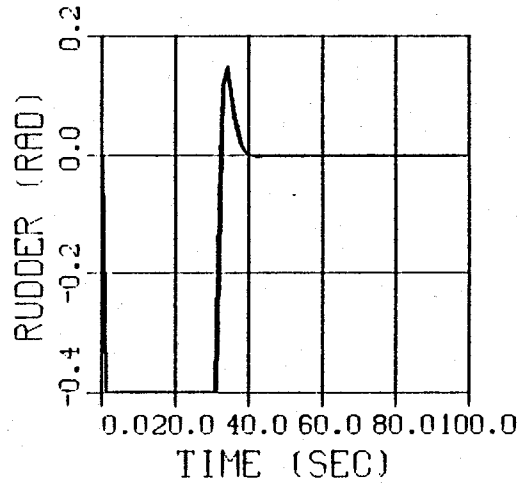
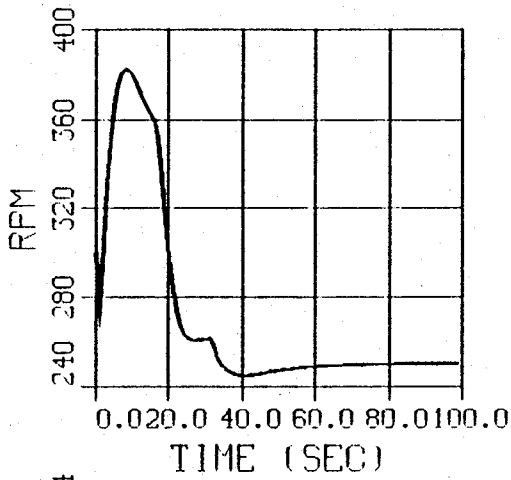
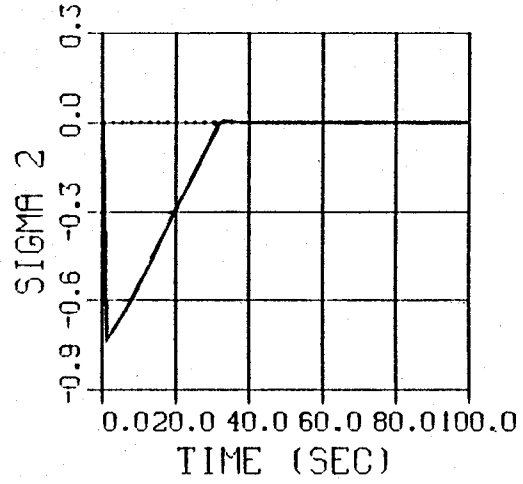
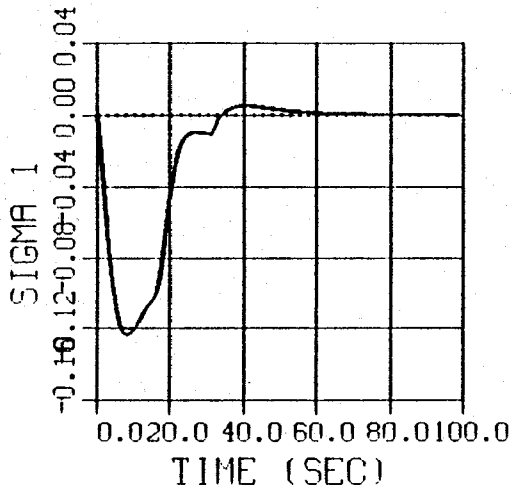


Figure 34. Run 3, Slow Speed Maneuver (3 ft/sec)

4. Conclusion

The autopilot has been verified to have the robustness necessary to operate within a speed range of 3 ft/sec to 50 ft/sec. The integration time step can be set at .005 sec for all values of operating speed or it can be related to a constant nondimensional time step of .1 using Equation 6.1. The advantage of varying the time step is that the calculations are performed faster for the slow speed runs where the time step can have a larger value.

C. LOS GUIDANCE SIMULATIONS

The final link in establishing that the AUV is ready for high speed operations is to show that LOS Guidance can still steer it through the desired way points at a speed of 50 ft/sec. This will be demonstrated by commanding the same basic maneuver performed in Chapter 5. To decrease the time needed to accelerate to the new higher speeds, the simulation will start with the AUV already at 50 ft/sec when the first way point is provided. The way points themselves will be further apart as is appropriate for high speed travel.

1. Run 4

The conditions for Run 4 were as follows:

- Commanded speed 50 Ft/sec.
- Integration time step .005 Sec.
- Target radius 2 Shiplengths
- Navigation update time .005 Sec.

The results of the run are shown in Figure 35. Though the AUV passed through all of the way points, the control of speed is not satisfactory. In exploring the speed control law for SISO1, it was determined that the nonlinear switching term was not strong enough to call for the necessary motor rpm to maintain the speed characteristics on the sliding surface.

2. Run 5

For Run 5 the value of the coefficient of the switching term was increased to 4000 thus enabling the switching term itself to call for enough rpm to result in a 50 ft/sec speed. With this modification, the speed control law for SISO1 becomes:

$$RPM = -4000.0 \text{ sat}ag1 + 83.33 u \quad (6.2)$$

With this new speed control law in place, the same maneuver was again commanded. The results are shown in Figure 36. The control of speed is now much better and the AUV still passes through the desired way points. Notice that when the desired speed is not attained, the control now calls for the maximum effort, which is 5000 rpm, from the propulsion motor. In order to improve speed control any further, a larger propulsion system would have to be utilized.

3. Run 6

This run explores the maximum Navigation update time interval that can be tolerated at high speeds before a decrease in performance is detected. Various runs were conducted with the first signs of trouble occurring when the navigation update time was 1 sec. The results for this run are shown in Figure 37. The need for a shorter Navigation

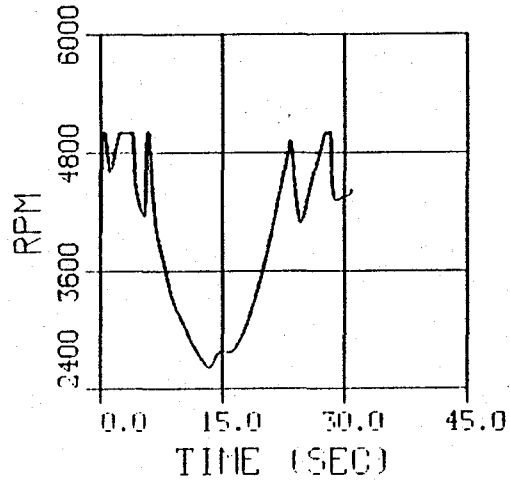
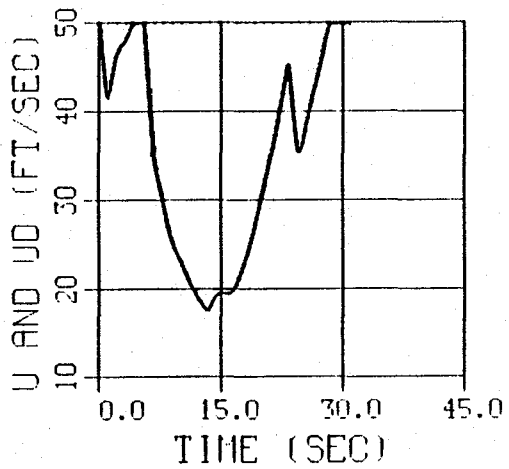
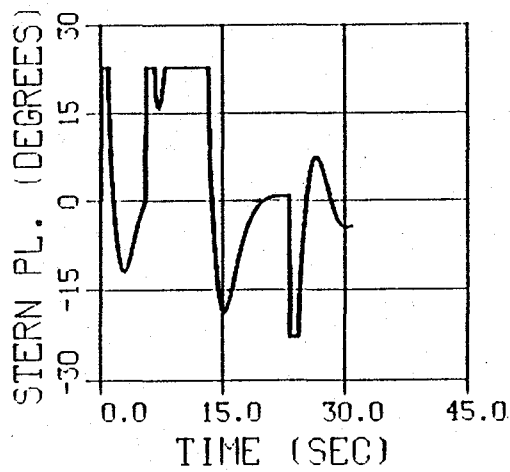
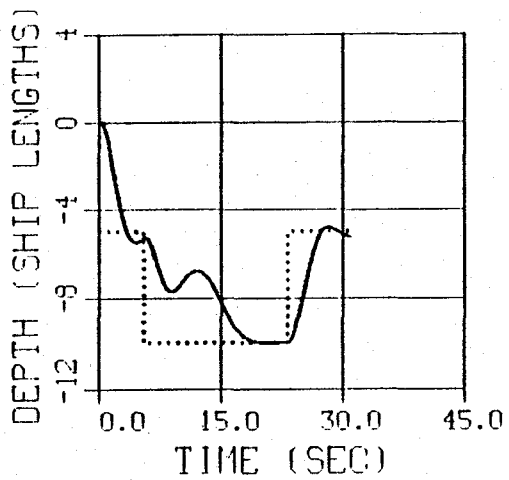
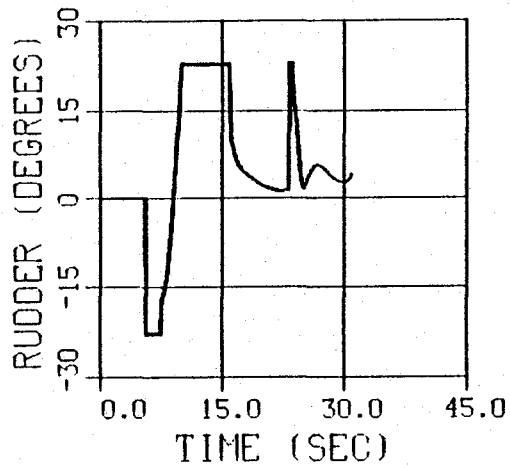
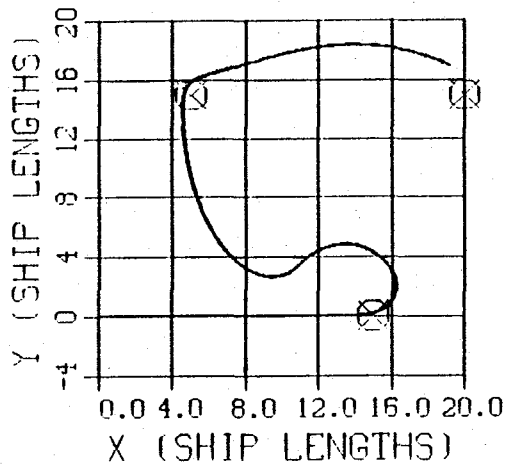


Figure 35. Run 4, High Speed, Multiple Way Point Maneuver (Initial Control Law)

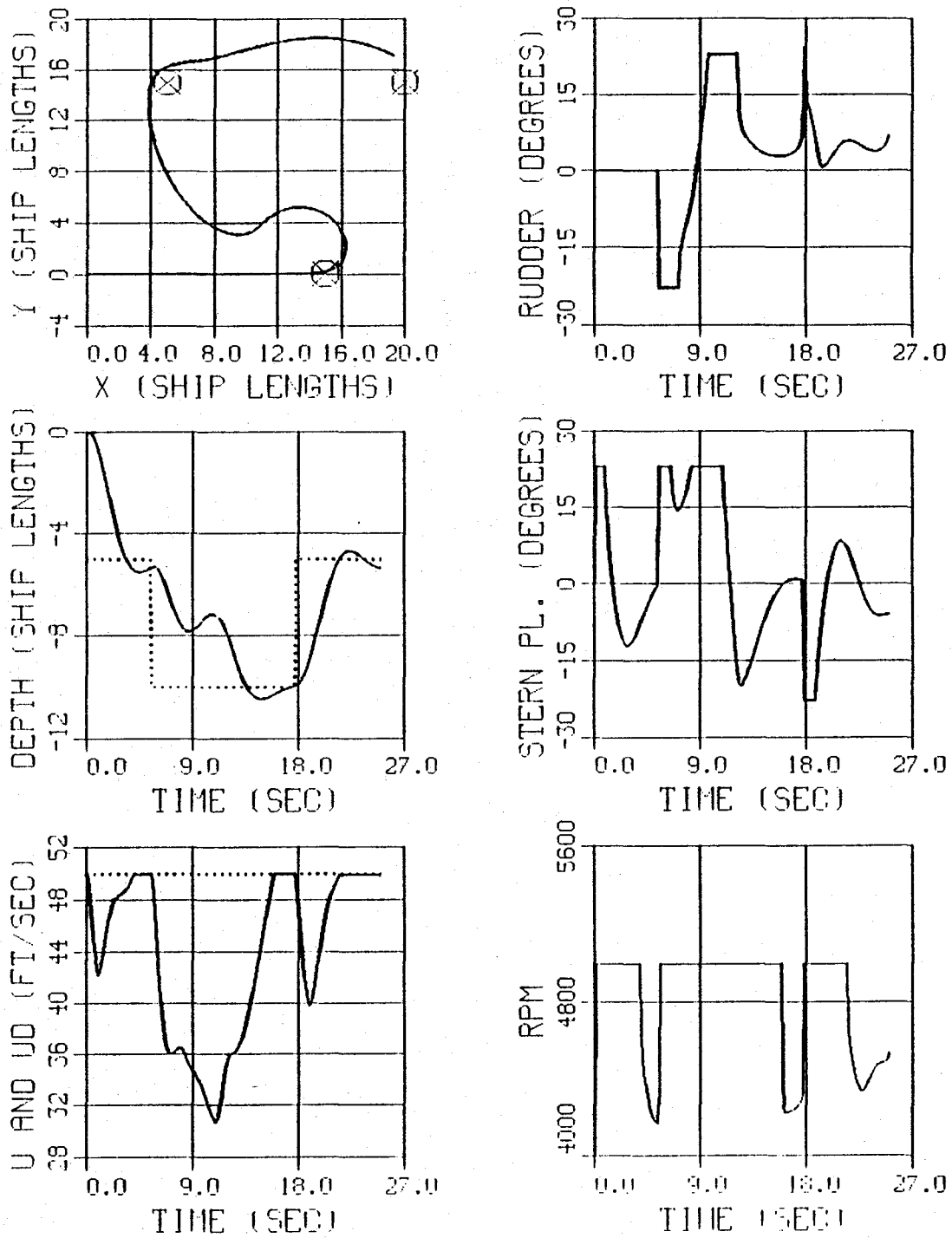


Figure 36. Run 5, High Speed, Multiple Way Point Maneuver (New Control Law)

update time is to be expected since the AUV travels further at high speeds during the update interval thus causing it to be further away from the last known position used by the control law.

4. Conclusion

LOS Guidance works as well at 50 ft/sec as it did at 3 ft/sec for the runs shown in Chapter 5. The only modifications that were required were to increase the strength of the switching term in the speed control law and decrease the integration time step and navigation update time. The fact that only the speed control law had to be modified is consistent with the fact that the control surfaces become more effective at higher speeds resulting in more response to a given control signal. To increase the speed, on the other hand, requires more motor rpm thus causing the speed control law to be modified consistent with the larger propulsion system that would have to be installed to achieve the higher operating speed.

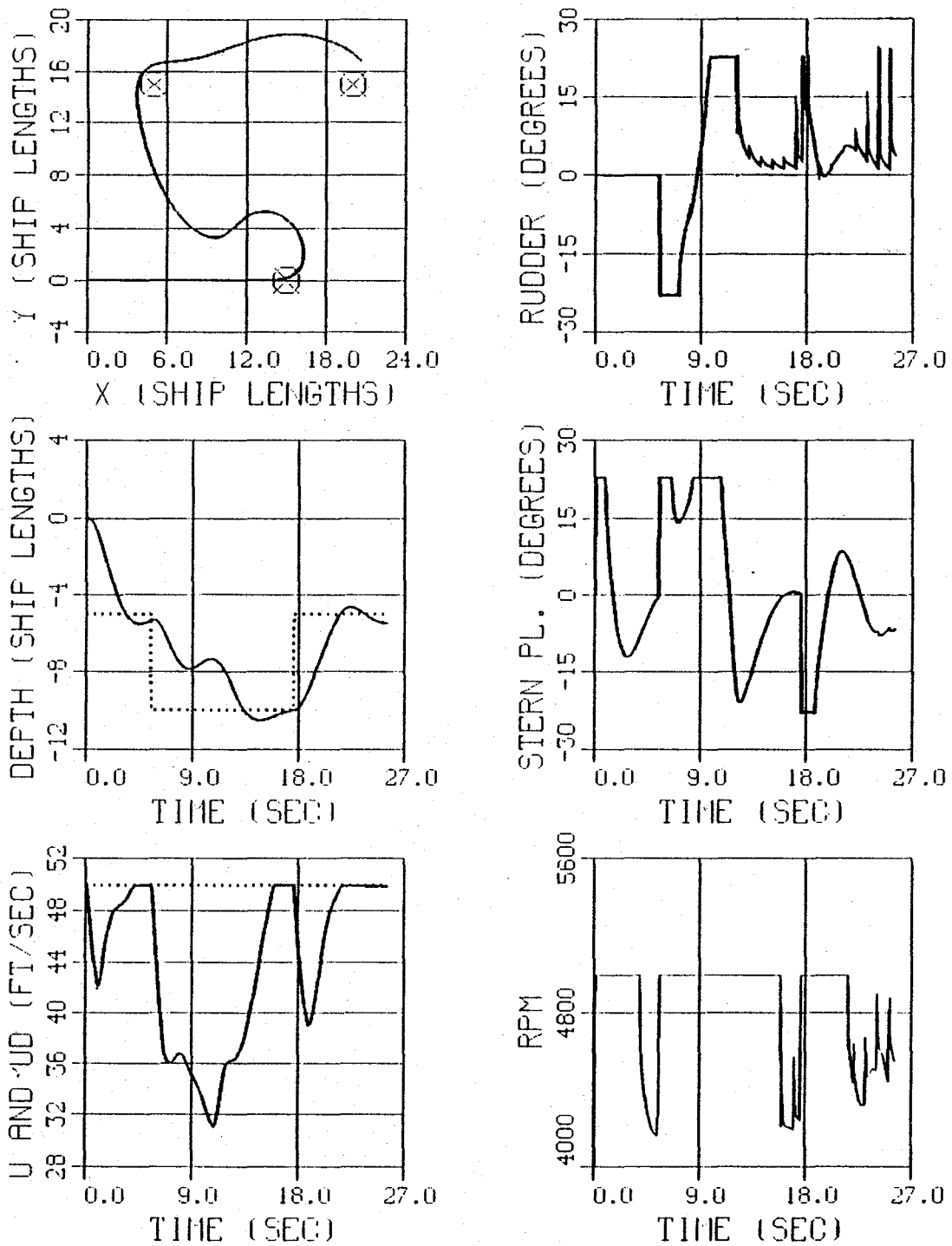


Figure 37. Run 6, Effect of Increasing Nav Update Time

VII. EFFECT OF CURRENT ON LOS GUIDANCE

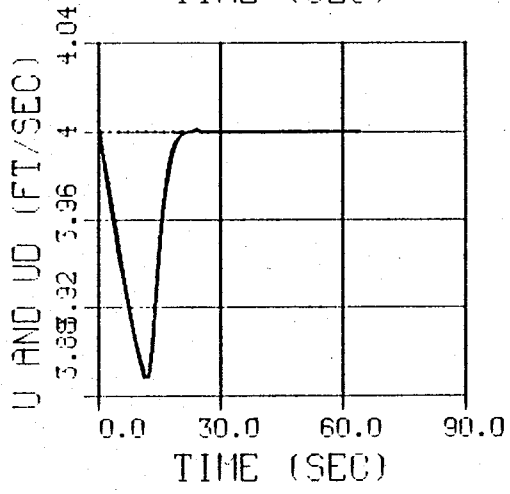
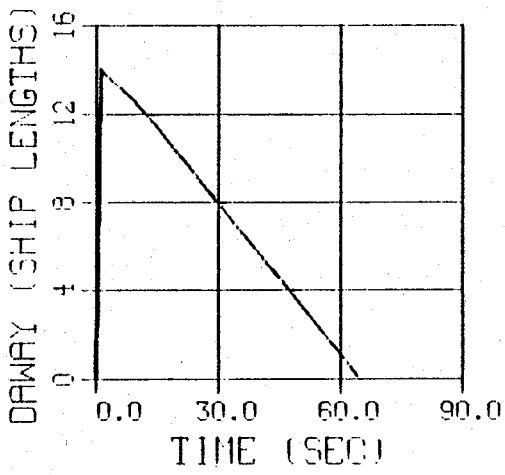
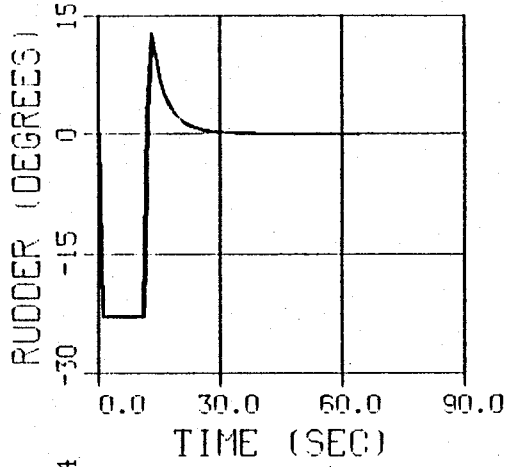
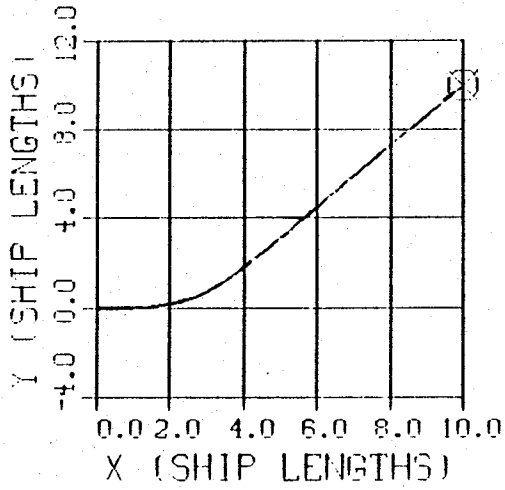
A. INTRODUCTION

This chapter investigates the effect of current on the autopilot's ability to reach a series of way points using LOS Guidance. The speed during these runs will be 4 ft/sec which is consistent with the operating speed of the AUV under construction at NPS. The first runs attempt to reach a single way point with the current coming from various directions. The next simulations will show the effect of the strength of the current. The last set of simulations show how the presence of a strong current can effect a path made up of multiple way points.

B. EFFECT OF CURRENT DIRECTION

1. Run 1

To see just the effect of the direction of the current, a single way point at (10,10,0) was chosen and a constant speed of 4 ft/sec was commanded for the AUV. Figure 38 shows the path that resulted when no current was present. Notice that a new graph of the distance the AUV is away from the next way point (DAWAY) has been added. The target radius was set at .01 shiplengths to allow the vehicle to get as close as possible to the prescribed way point. This represents the standard to which all other runs can be compared.



**NO
CURRENT**

Figure 38. Reaching a Single Way Point With No Current

2. Run 2 and Run 3

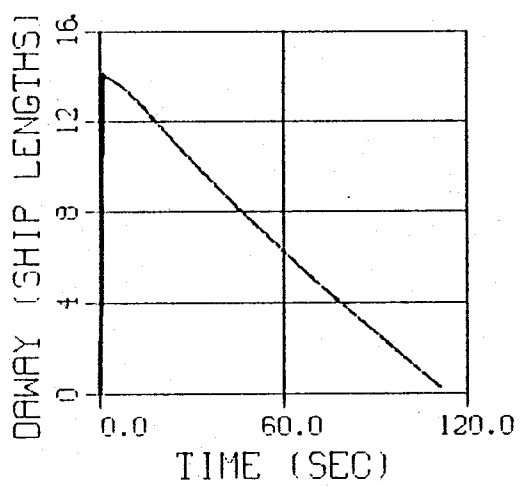
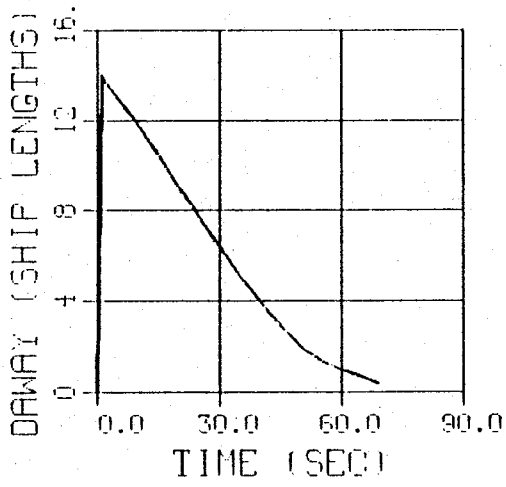
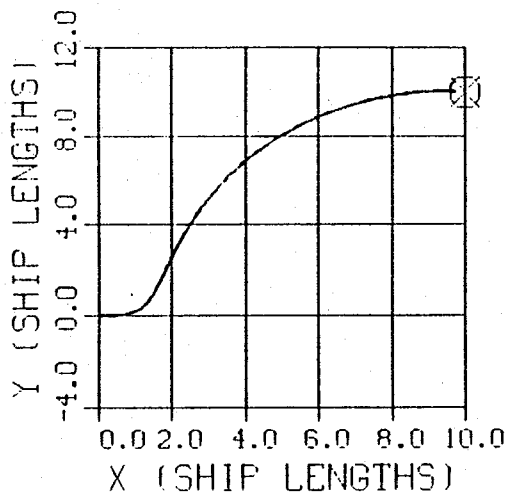
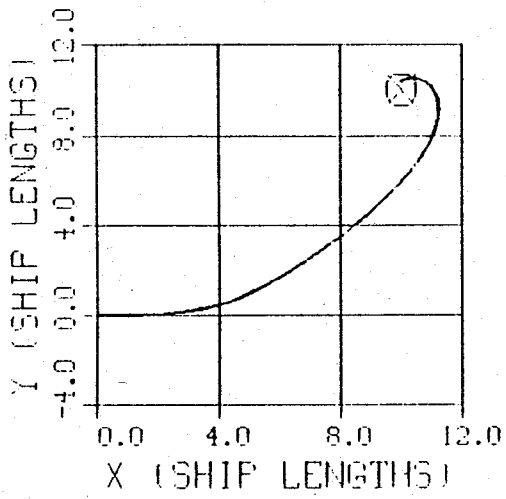
These runs, shown in Figure 39, demonstrate the effect of a 2 ft/sec current. This current is 50 % of the operating speed of the AUV which should be considered to be an extreme case. Both runs are shown in Figure 39. For Run 2 the current was from left to right and for Run 3 the current was right to left. Notice that the AUV still reaches the way point in both cases but is swept away from the no current path. Notice too that in both cases the AUV has a final approach which is against the current.

3. Run 4 and Run 5

For Run 4 the current direction was from top to bottom. For Run 5 it was from bottom to top. In both cases the final approach once again is against the current, see Figure 40. In addition the miss distance was the largest when the current comes from bottom to top. It should be remembered however, that the target radius that will be used during the actual operation is two shiplengths. Since the AUV was able to get within approximately 1.5 shiplengths, this run should be considered a hit also.

4. Conclusion

The AUV will reach the prescribed way point under extremely adverse current conditions. The problem however is that its path deviates from the near straight line path of the no current case to the point that it may be unsuitable for operation in an environment which requires a high degree of control over the path. The other problem to address is that the AUV will approach the way point against the current which may be undesirable considering the position of the next way point along its path.



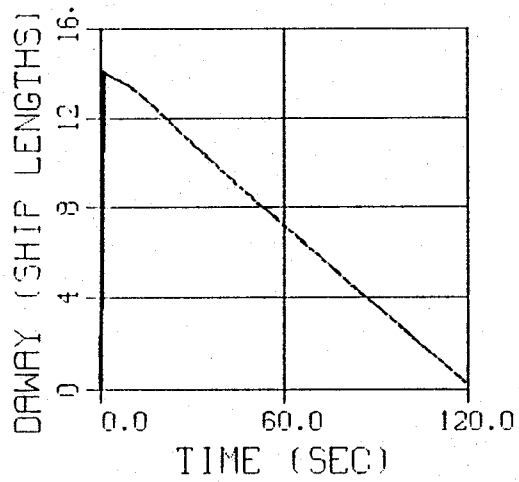
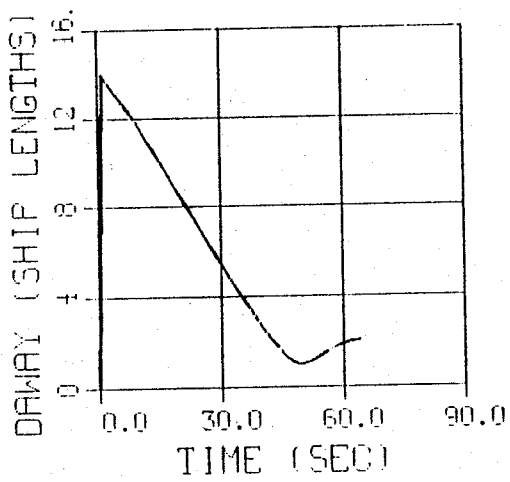
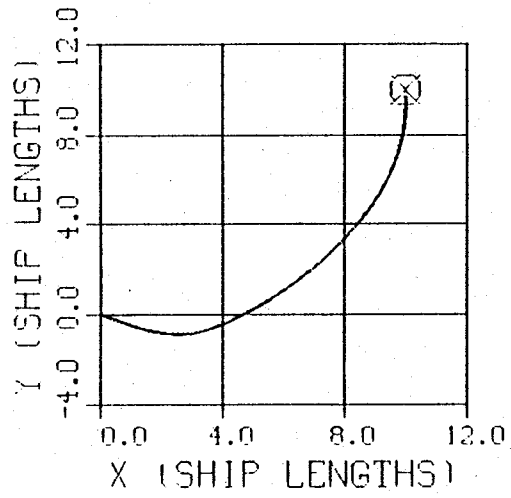
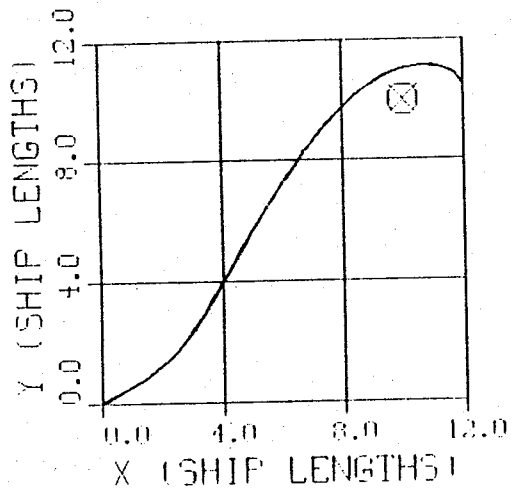
CURRENT

→

CURRENT

←

Figure 39. Effect of a 2 FPS Current in the X Direction



CURRENT ↑

CURRENT ↓

Figure 40. Effect of a 2 FPS Current in the Y Direction

C. ADAPTATION OF STEERING CONTROL

For the runs shown in Figures 39 and 40 the value for the x and y position of the AUV are calculated by the following equations where the subscript c denotes current.

$$\begin{aligned}\dot{x} &= u_c + u \cos\psi - v \sin\psi \\ \dot{y} &= v_c + u \sin\psi + v \cos\psi\end{aligned}\tag{7.1}$$

Recall too that the desired heading is calculated using Equation 7.2.

$$\psi_d = \text{ATAN}[(y-y_d) / (x-x_d)]\tag{7.2}$$

Recall that at steady state the desired heading and the actual heading are equal. Therefore, at steady state, with a desired way point at (0,0) for simplicity, Equation (7.2) can be substituted into Equation (7.1) which upon integration yields the result shown as Equation (7.3).

$$\psi_d = \text{ATAN}(v_c/u_c)\tag{7.3}$$

Equation (7.3) shows that the desired heading at steady state ultimately is parallel to the direction of the current. This result is verified by the simulation results shown in Figures 39 and 40. Notice that if the vertical component of the current is zero as is the case for Figure 39 the desired heading becomes 0 or 180 degrees. Correspondingly, if the horizontal component of the current is zero, the desired heading is + or - 90 degrees as shown in Figure 40.

Once an understanding of the above relationships was attained, the next step was to modify the control law to attain tighter control over path. This was done by specifying a desired heading that compensates for the direction of the current rather than being controlled by it. To do this, recall that the equation for the sliding surface for SISO2 is given by Equation (2.16) which is repeated here as follows:

$$\sigma = s_1(\psi - \psi_d) + s_2(v) + s_3(r) \quad (7.4)$$

By considering the simple case where the AUV is commanded to travel along the x axis, a modification to the sliding surface can be attained. For this simple path the desired heading is zero. In addition the speed in the y direction must also be zero. This observation coupled with Equation (7.1) gives the relationship shown in Equation (7.5) that can be used to find the steady state error that must exist to compensate for the effect of the current.

$$\psi_s = -ASIN(v_c/u) \quad (7.5)$$

This result requires that the actual heading of the AUV be composed of two parts as shown in Equation (7.6).

$$\psi = \psi_d + \psi_s \quad (7.6)$$

Recalling that the sway velocity v has been estimated to be zero and that r is zero at

steady state, the sliding surface equation shown in Equation (7.4) becomes:

$$\sigma = s_1 \psi = s_1 \psi_s$$

Notice that the sliding surface equation will not be zero at steady state unless another term equal in magnitude but opposite in sign is added to the sliding surface equation. It is by this reasoning that the equation for the sliding surface for SISO2 was modified to become:

$$\sigma = s_1 (\psi - \psi_d) + s_2 v + s_3 r + s_1 A \sin(v_c/u)$$

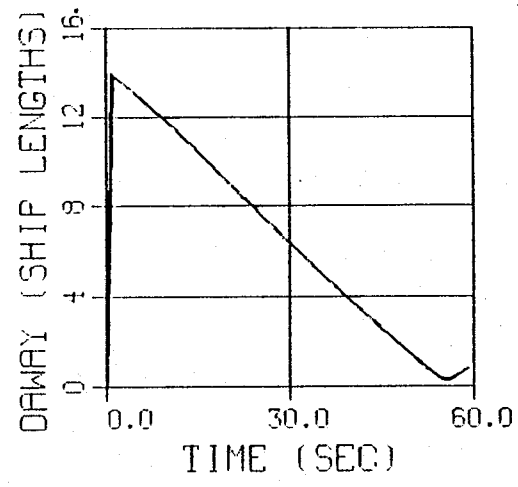
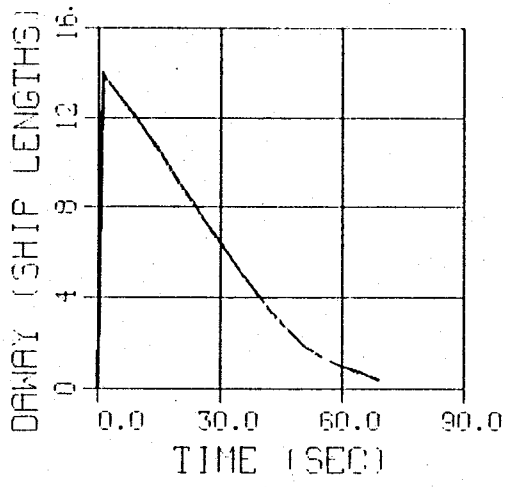
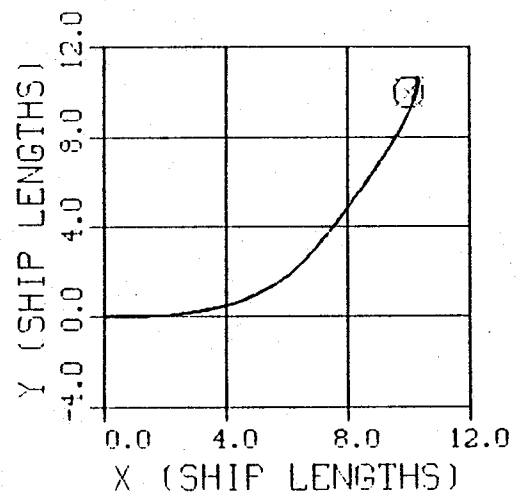
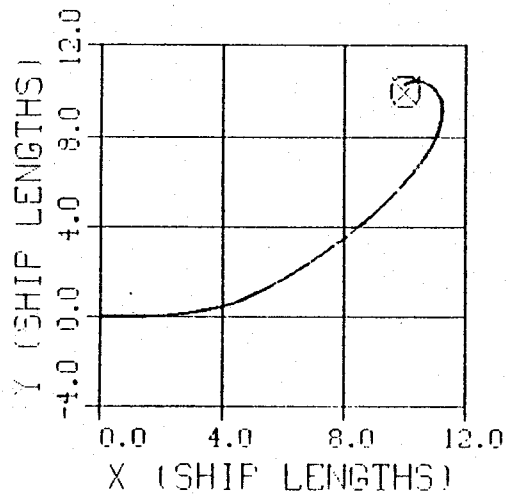
This additional term has the effect of introducing a steady state error for heading that exactly offsets the effects of the current and improves the path keeping characteristics of the AUV. It should be pointed out that this equation relies on perfect knowledge of the current velocity allowing the component of the current's velocity that is perpendicular to the AUV's straight line path between consecutive way points to be determined.

The improvement which results is shown by rerunning the same simulations discussed at the beginning of this chapter using the modified sliding surface for SISO2. The improvement is easily seen in Figures 41,42,43, and 44 where the old results are on the left and the new results are on the right. Based upon this comparison, it can be concluded that LOS Guidance can be highly successful at controlling an AUV in a high current environment if the velocity of the current is well known. This shifts the burden of tight control over the path to the design of an accurate estimator for current velocity.

Such an observer will not be included in this work but the following simulation is included to give an indication of the impact an inaccurate estimate of current would have. The current was given a value of 2 ft/sec from bottom to top. The estimated value was 1 ft/sec from bottom to top. Figure 45 shows that the result is still improved from the case where no compensation for current was included. From this it can be concluded that even an imperfect estimate of the currents velocity will result in some improvement in the AUV's path. The better the estimate the more improvement that results.

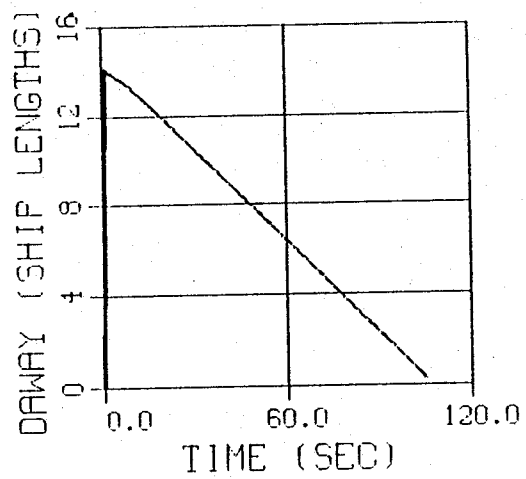
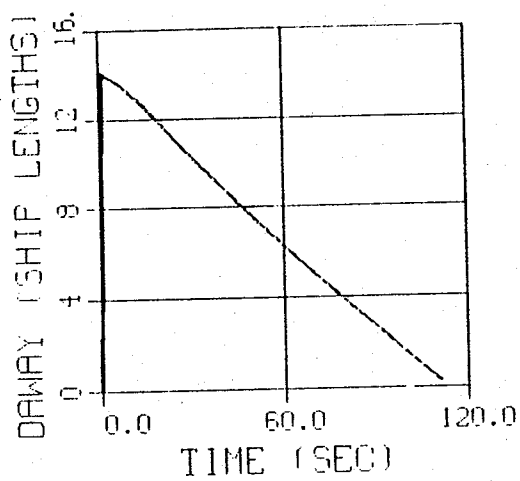
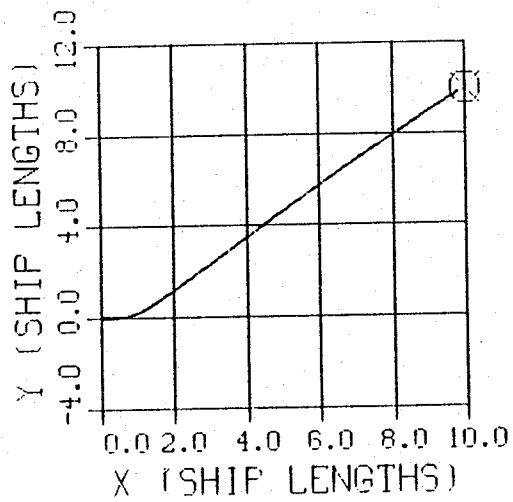
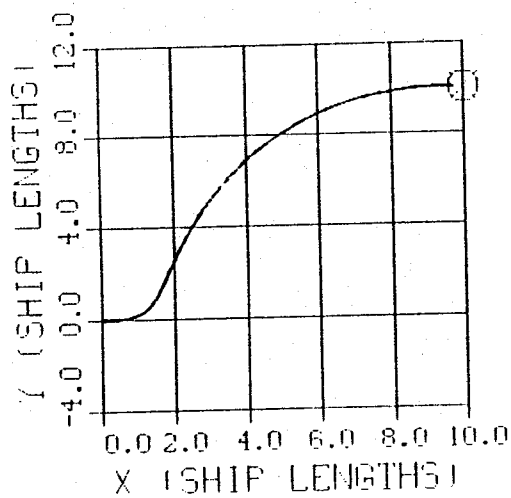
D. MULTIPLE WAY POINT PATHS

The ultimate test for the autopilot using LOS Guidance is to show that it can reach a series of way points. The same path used through out this work is repeated here one last time to demonstrate that the AUV will travel through a series of way points even in the presence of a current that has a magnitude of 1.4 ft/sec in a direction 45 degrees away from the origin. The result is shown in Figure 46 which can be compared to the same run done without a current present in Figure 47. The results show that LOS Guidance based on a heading control autopilot that has been modified to account for the presence of a current is very capable of guiding the AUV through the prescribed way points.



CURRENT

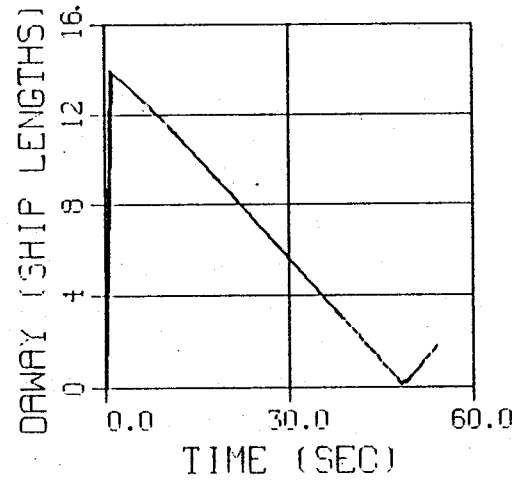
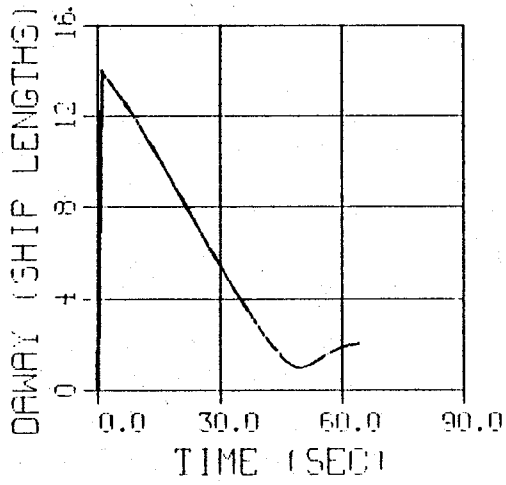
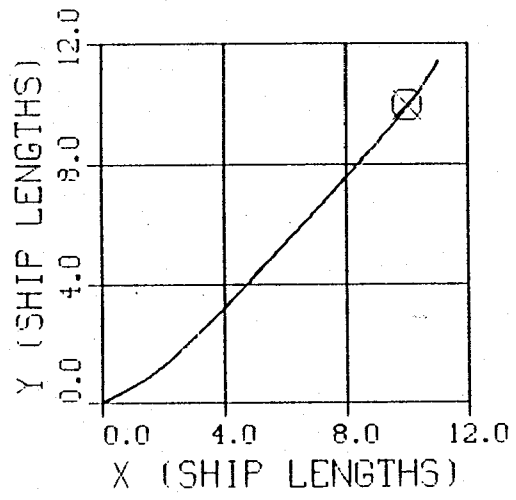
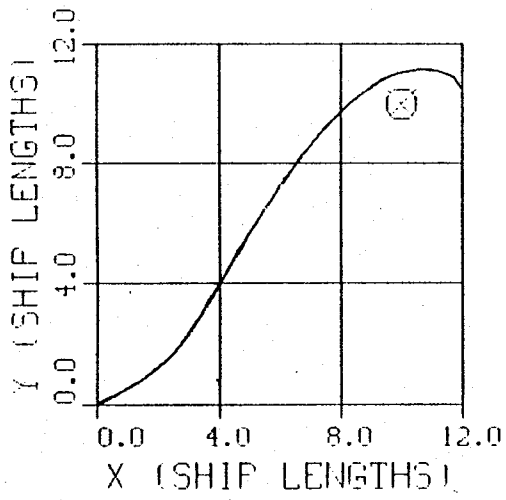
Figure 41. Original Autopilot (Left) vs Modified (Right)



CURRENT

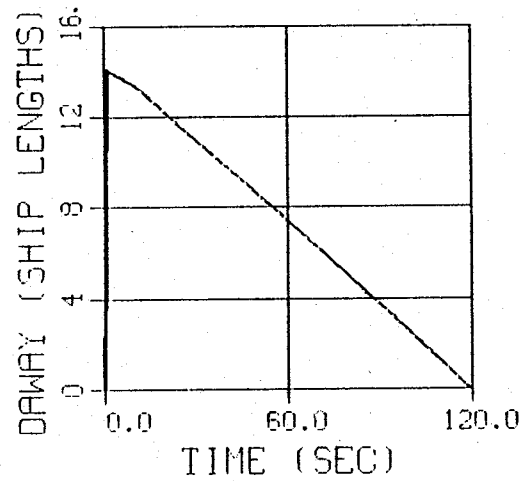
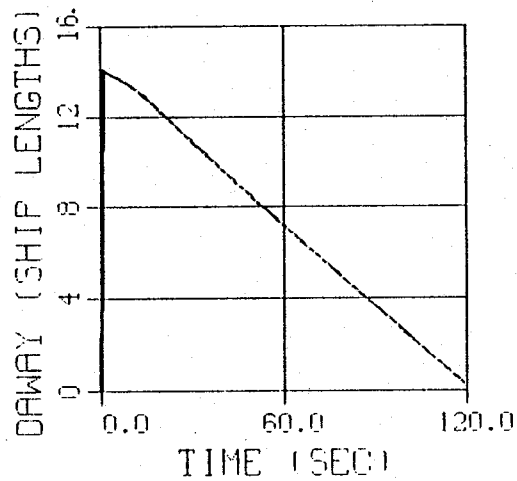
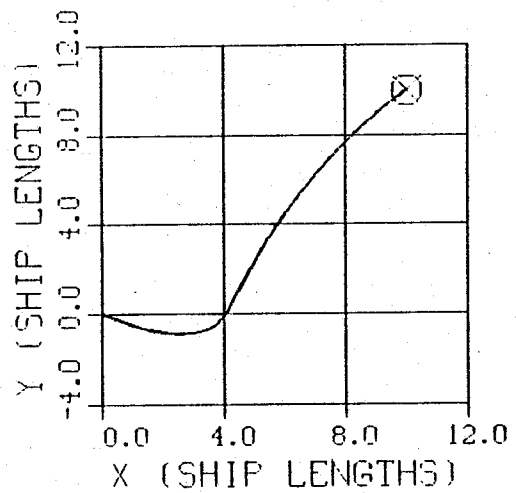
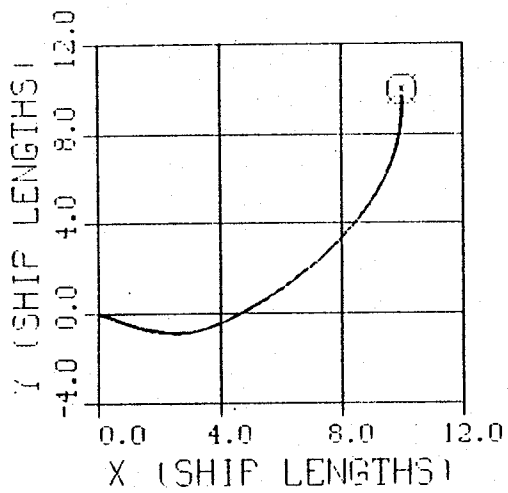


Figure 42. Original Autopilot (Left) vs Modified (Right)



CURRENT ↑

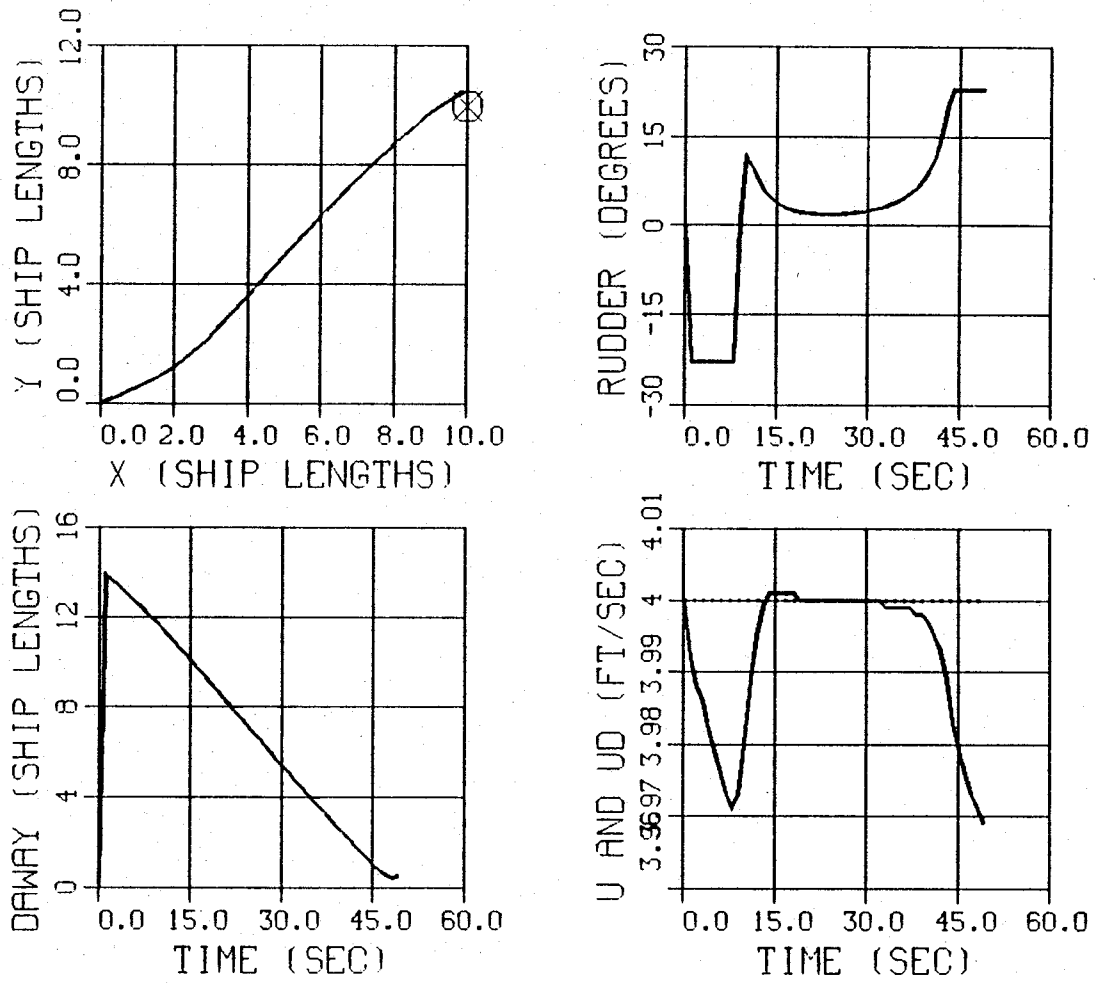
Figure 43. Original Autopilot (Left) vs Modified (Right)



CURRENT

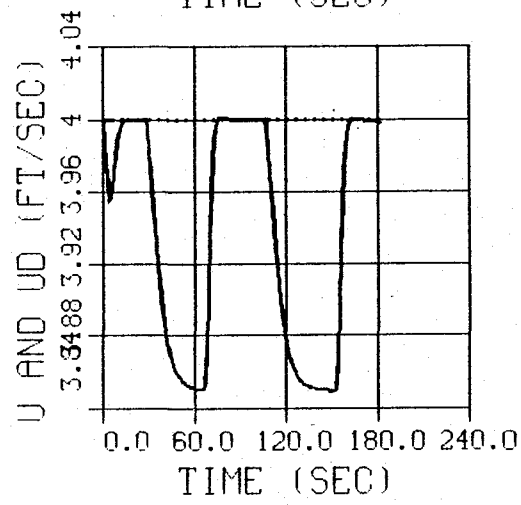
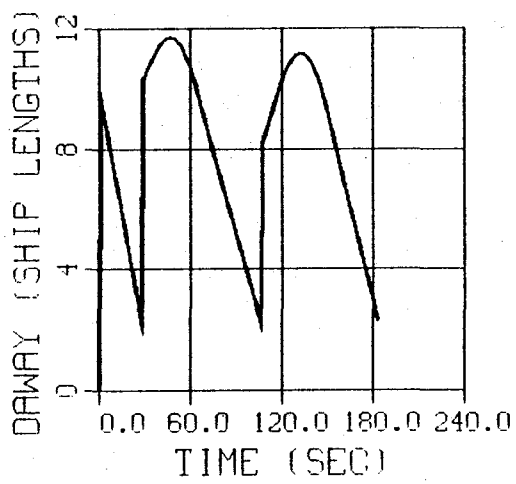
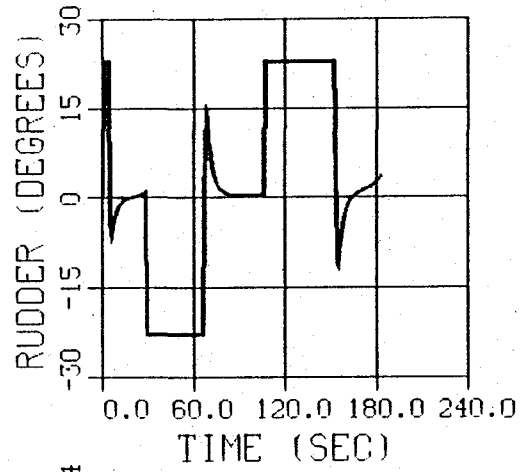
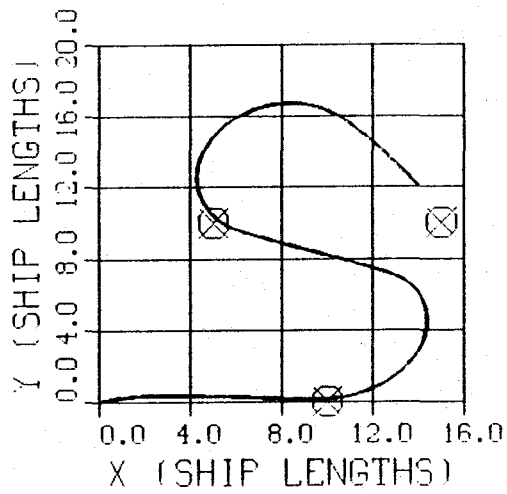


Figure 44. Original Autopilot (Left) vs Modified (Right)



CURRENT ↑

Figure 45. Effect of an Inaccurate Current Velocity Estimate



CURRENT

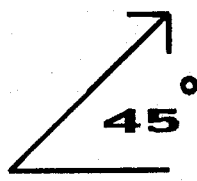
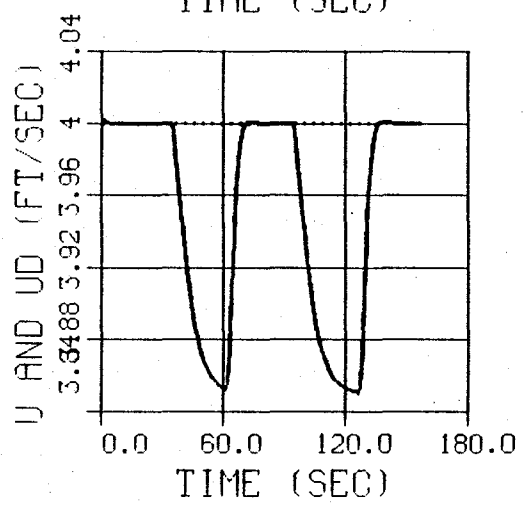
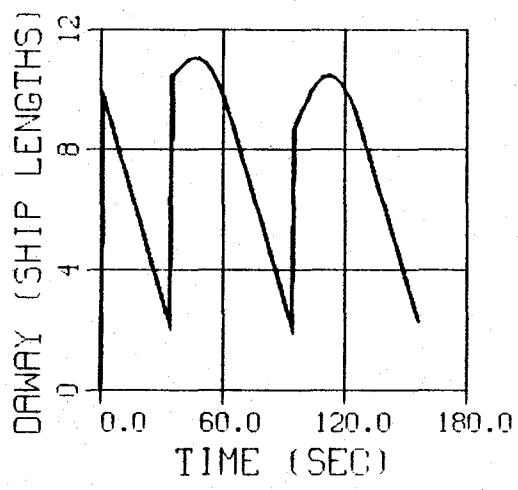
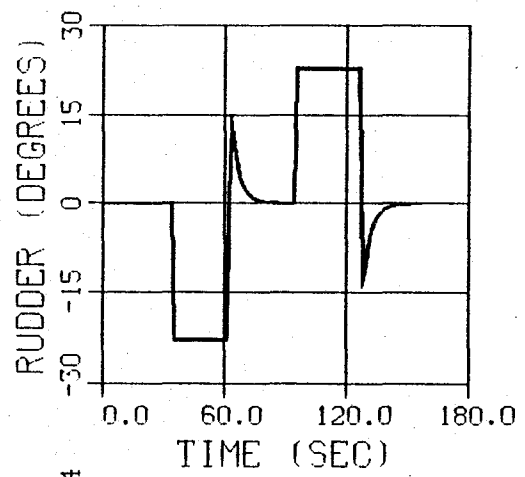
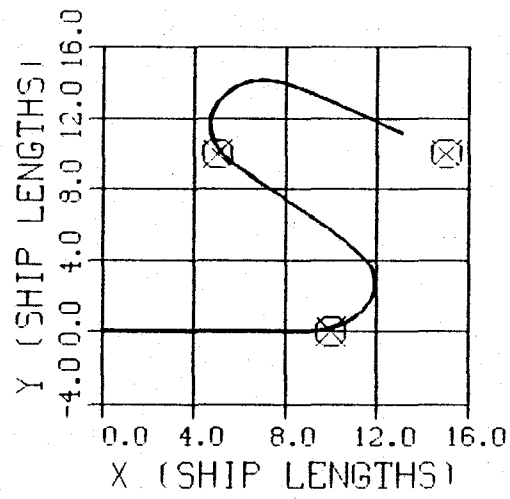


Figure 46. Traveling to Way Points in a Strong Current



**NO
CURRENT**

Figure 47. Traveling to Way Points (No Current Case)

VIII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The following are the major conclusions that can be drawn from this study.

- The control laws for speed, heading, and depth were all developed independently, and yet their individual control actions did not interfere with each other.
- The control laws, based on a simple linear model, were able to control the more complex nonlinear model of the SDV MARK 9 vehicle by virtue of the nonlinear switching terms, associated with Sliding Mode Theory, that compensated for any unmodeled behavior and enhanced robustness.
- In developing the steering control autopilot, sway velocity is best handled as a disturbance rather than an observed quantity.
- LOS Guidance provides an effective means of interpreting the position coordinates of the way points provided by the mission planner into commands for heading that can be executed by the autopilot. An inherent characteristic of LOS Guidance is that it enhances the robustness of the system as a whole. In addition, tighter control of path can be attained by specifying the way points more frequently.
- The robustness of this combination of Sliding Mode Control and LOS Guidance is demonstrated by the 3 ft/sec to 50 ft/sec speed envelope over which the autopilot operated effectively even though the linearization that produced the simple linear model upon which the control laws were based was done about a nominal operating speed of 6 ft/sec.
- LOS Guidance can be easily modified to take advantage of known or observed current velocity thus allowing the vehicle to operate in strong currents.

B. RECOMMENDATIONS

The following actions should be taken to fully utilize the research done for this thesis.

- Verify simulation results using the actual AUV currently under construction at NPS.
- Contrast this autopilot using LOS Guidance to other techniques for controlling AUVs, such as minimizing cross track error, to ensure the best method is ultimately adopted.
- Design an observer that can determine the velocity of the current that the AUV is operating in.


```

// *****
// *
// * CALCULATION OF A AND B MATRICIES FOR SISO1 WHERE X=[u]' *
// *
// *****

```

```

A1 = -ALPHA*ubar;
B1 = ALPHA*.012**2*nbar;

```

```

// *****
// *          DECOUPLING THE LATERAL AND YAW EQUATIONS OF MOTION *
// *****
M=0.0*ones(2,2);
C=0.0*ones(2,3);

```

```

M(1,1)=mass-den*L**3*Yvd/2;
M(1,2)=-den*L**4*Yrd/2;
M(2,1)=-den*L**4*Nvd/2;
M(2,2)=Iz-den*L**5*Nrd/2;
//C(1,1)=den*L**3*Yr*ubar/2;
C(1,1)=den*L**3*Yr*ubar/2 - mass*ubar;
C(1,2)=den*L**2*Yv*ubar/2;
C(1,3)=den*L**2*Ydr*ubar**2/2;
C(2,1)=den*L**4*Nr*ubar/2;
C(2,2)=den*L**3*Nv*ubar/2;
C(2,3)=den*L**3*Ndr*ubar**2/2;

```

```

// *****
// *
// * CALCULATION OF A AND B MATRICIES FOR SISO2 WHERE X=[Heading,v,r]' *
// *
// *****

```

```

D=inv(M)*C;
A2=[0,0,1;0,d(1,2),d(1,1);0,d(2,2),d(2,1)];
B2=[0;d(1,3);d(2,3)];

```

```

// ***** CALCULATION OF S1,S2,S3,S4 *****

```

```

inquire atasq1
inquire atasq2

inquire poles2

kc2= poleplace(A2,B2,poles2);
AC2=A2-B2*kc2;

[evect2,eval2]=eig(AC2');

S1=1;
S2=evect2(1,1);
S3=evect2(2,1);
S4=evect2(3,1);

```

```

// *****
// *      DETERMINING GAIN VALUES FOR MATRIXx BLOCK DIAGRAM.      *
// *****
// CONVENIENT GROUPING OF TERMS:
DD1=COUD*.012**2*nbar;
DD2=S3*d(1,3)+S4*d(2,3);
Nata1=-atasq1/DD1;
Nata2=-atasq2/DD2;
NN1=S1*A1/DD1;
NN2=(S3*d(1,2) + S4*d(2,2))/DD2;
NN3=(S2+ S3*d(1,1) + S4*d(2,1))/DD2;

d23=d(2,3);
d21=d(2,1);
d13=d(1,3);
d22=d(2,2);
d12=d(1,2);
d11=d(1,1);
// *****
// *                               SIMULATION                               *
// *****
inquire tsim
tstep=.1;
inquire ud
inquire HdDEG
Hd = HdDEG*6.28/360;
T=[0:tstep:tsim]';
ud=ud*ones(T);
Hd=Hd*ones(T);
U=[ud,Hd];
Y = SIM(T,U);
// *****
// *          GRAPHICAL DISPLAY OF RESULTS          *
// *****
HHD=[Y(:,2)*360/6.28 Hd*360/6.28];
UUD=[Y(:,1) ud];

PLOT(T,Y(:,3),'UPPER LEFT XLABEL/T/ YLABEL/SIGMA1/TITLE/ ATASQ1=.6 ...
BAND=1FPS/')

PLOT(T,Y(:,5),'UPPER RIGHT XLABEL/T/ YLABEL/SIGMA2/TITLE/ATASQ2=.4 POLES...
=[0,-.4,-.42] BAND=.1RAD./')

PLOT(T,Y(:,6),'LOWER RIGHT XLABEL/T/ YLABEL/RUDDER/TITLE/ STEERING ...
(MAX DEFLECTION=.4RAD)/')

PLOT(T,Y(:,4),'LOWER LEFT XLABEL/T/ YLABEL/RPM/TITLE/ PROPULSION...
(MAX RPM= 500RPM)/')

PLOT(T,HHD,'UPPER RIGHT XLABEL/T/ YLABEL/H & HD/TITLE/ HEADING ...
(Hd = 90 DEG)/')
PLOT(T,UUD,'UPPER LEFT XLABEL/T/ YLABEL/ U & UD/TITLE/ SPEED ...
(Ud = 4.0 FPS)/')
//exec('RUNSUPERG1.X')

```

APPENDIX B.

```

C *****
C *
C *   NONLINEAR AUV MODEL / STERN PLANE AND BOW PLANE SEPARATED *
C *
C *   VARIABLE DECLARATION 11           INPUTS           280 *
C *   CONSTANTS           80           PROPULSION MODEL 300 *
C *   INITIAL CONDITIONS  136          OUTPUTS          452 *
C *   MASS MATRIX         230          INTEGRATION       475 *
C *   INVERT MATRIX       268          CONTROL LAWS     500 *
C *****
REAL AW(82,82)
REAL MASS,LATYAW,NORPIT
REAL MM(6,6),G4(4),GK4(4),BR(4),HH(4)
REAL B(6,6),BB(6,6)
REAL A(12,12),AA(12,12),INDX(100)
REAL XPP ,XQQ ,XRR ,XPR
REAL XUDOT ,XWQ ,XVP ,XVR
REAL XQDS ,XQDB ,XRDR ,XVV
REAL XWW ,XVDR ,XWDS ,XWDB
REAL XDSDS,XDBDB ,XDRDR ,XQDSN
REAL XWDSN ,XDSDSN
REAL TIME,S,EITA,UBAR,UHAT,COMZ,BAR,SIM,DE,SAT,VHAT,ZZOBS
REAL SS1,SS2,UD,XD,YD,TD,TNWP,XA,YA,HD,HDMDEG,DAWAY,SATSGN1
REAL SATSGN2,ZZOBSDOT,NEARMISS,ALPHA,VCC,XD1,YD1,XD2,YD2,DY,DX
REAL NAVUPDATE,TNAV,TARGET,FF,GG,HHH,LLL,HDP,HDM,LDAWAY
INTEGER DV

C
C   LATERAL HYDRODYNAMIC COEFFICIENTS
C
REAL YPDOT ,YRDOT,YPQ ,YQR
REAL YVDOT ,YP ,YR ,YVQ
REAL YWP ,YWR ,YV ,YVW
REAL YDR ,CDY

C
C   NORMAL HYDRODYNAMIC COEFFICIENTS
C
REAL ZQDOT ,ZPP,ZPR ,ZRR
REAL ZWDOT ,ZQ ,ZVP ,ZVR
REAL ZW ,ZVV ,ZDS ,ZDB
REAL ZQN ,ZWN ,ZDSN ,CDZ
REAL ZHADOT,ZHAT

C
C   ROLL HYDRODYNAMIC COEFFICIENTS
C
REAL KPDOT ,KRDOT ,KPQ ,KQR
REAL KVDOT ,KP ,KR ,KVQ
REAL KWP ,KWR ,KV ,KVW
REAL KPN ,KDB

C
C   PITCH HYDRODYNAMIC COEFFICIENTS
C
REAL MQDOT ,MPP ,MPR,MRR
REAL MWDOT ,MQ ,MVP ,MVR
REAL MW ,MVV ,MDS ,MDB
REAL MQN ,MWN ,MDSN

```

REAL QHADOT,QHAT,THADOT,THAT

YAW HYDRODYNAMIC COEFFICIENTS

REAL NPDOT,NRDOT,NPQ ,NQR
REAL NVDOT , NP ,NR ,NVQ
REAL NWP , NWR ,NV ,NVW
REAL NDR

MASS CHARACTERISTICS OF THE FLOODED VEHICLE

REAL WEIGHT , BOY ,VOL ,XG
REAL YG , ZG ,XB ,ZB
REAL IX , IY ,IZ ,IXZ
REAL IYZ , IXY ,YB
REAL L , RHO ,G ,NU
REAL A0 ,KPROP ,NPROP , X1TEST
REAL DEGRUD ,DEGSTN
COMMON /BLOCK1/ F(12), FP(6), XMMINV(6,6), UCF(4)
INTEGER N,IA,IDGT,IER,LAST,J,K,M,JJ,KK,I
REAL WKAREA(54), X(12)

RUDDER COEFFICIENTS

PARAMETER (DSMAX= -0.175)

LONGITUDINAL HYDRODYNAMIC COEFFICIENTS

PARAMETER(XPP = 7.E-3 ,XQQ = -1.5E-2 ,XRR = 4.E-3 ,XPR =7.5E-4,
& XUDOT=-7.6E-3 ,XWQ = -2.E-1 ,XVP = -3.E-3 ,XVR = 2.E-2,
& XQDS=2.5E-2 ,XQDB=-2.6E-3 ,XRDR= -1.E-3 ,XVV =5.3E-2,
& XWW =1.7E-1 ,XVDR=1.7E-3 ,XWDS=4.6E-2 ,XWDB= 1.E-2,
& XSDS= -1.E-2 ,XDBDB= -8.E-3 ,XDRDR= -1.E-2 ,XQDSN= 2.E-3,
& XWDSN=3.5E-3 ,XSDSN= -1.6E-3)

LATERAL HYDRODYNAMIC COEFFICIENTS

PARAMETER(YPDOT=1.2E-4 ,YRDOT=1.2E-3 ,YPQ = 4.E-3 ,YQR =-6.5E-3,
& YVDOT=-5.5E-2 ,YP = 3.E-3 ,YR = 3.E-2 ,YVQ =2.4E-2,
& YWP =2.3E-1 ,YWR =-1.9E-2 ,YV = -1.E-1 ,YVW =6.8E-2,
& YDR =2.7E-2 ,CDY =3.5E-1)

NORMAL HYDRODYNAMIC COEFFICIENTS

PARAMETER(ZQDOT=-6.8E-3 ,ZPP =1.3E-4 ,ZPR =6.7E-3 ,ZRR =-7.4E-3,
& ZWDOT=-2.4E-1 ,ZQ =-1.4E-1 ,ZVP =-4.8E-2 ,ZVR =4.5E-2,
& ZW = -3.E-1 ,ZVV =-6.8E-2 ,ZDS =-7.3E-2 ,ZDB =-2.6E-2,
& ZQN =-2.9E-3 ,ZWN =-5.1E-3 ,ZDSN= -1.E-2 ,CDZ = 1.0)

ROLL HYDRODYNAMIC COEFFICIENTS

PARAMETER(KPDOT= -1.E-3 ,KRDOT=-3.4E-5 ,KPQ =-6.9E-5 ,KQR =1.7E-2,
& KVDOT=1.3E-4 , KP =-1.1E-2 ,KR =-8.4E-4 ,KVQ=-5.1E-3,
& KWP =-1.3E-4 , KWR =1.4E-2 ,KV =3.1E-3 ,KVW =-1.9E-1,
& KPN =-5.7E-4 , KDB = 0.0)

PITCH HYDRODYNAMIC COEFFICIENTS

PARAMETER(MQDOT=-1.7E-2 ,MPP =5.3E-5 ,MPR = 5.E-3 ,MRR =-2.9E-3,
& MWDOT=-6.8E-3 , MQ =-6.8E-2 ,MVP =1.2E-3 ,MVR =1.7E-2,

```
& MW = 1.E-1      , MVV =-2.6E-2   , MDS =-4.1E-2   , MDB =6.9E-3,
& MQN =-1.6E-3    , MWN =-2.9E-3   , MDSN =-5.2E-3)
```

```
C
C
C      YAW HYDRODYNAMIC COEFFICIENTS
```

```
PARAMETER(NPDOT=-3.4E-5 ,NRDOT=-3.4E-3,NPQ =-2.1E-2 ,NQR =2.7E-3,
& NVDOT=1.2E-3      , NP =-8.4E-4      ,NR =-1.6E-2      ,NVQ = -1.E-2,
& NWP =-1.7E-2      , NWR =7.4E-3       ,NV =-7.4E-3      ,NVW =-2.7E-2,
& NDR =-1.3E-2)
```

```
C
C
C      MASS CHARACTERISTICS OF THE FLOODED VEHICLE
```

```
PARAMETER( WEIGHT =12000., BOY =12000. ,VOL =200. ,XG = 0. ,
& YG = 0.0          , ZG = 0.20         ,XB = 0.          ,ZB = 0.0 ,
& IX = 1500.        , IY = 10000.        ,IZ = 10000.      ,IXZ = -10. ,
& IYZ = -10.        , IXY = -10.         ,YB = 0.0 ,
& L = 17.4          , RHO = 1.94         ,G = 32.2         ,NU = 8.47E-4 ,
& A0 = 2.0          ,KPROP = 0.          ,NPROP = 0. ,
& DEGRUD= 0.0       ,DEGSTN= 0.0)
```

```
C
C
C      INPUT INITIAL CONDITIONS HERE IF REQUIRED
```

```
OPEN(20,FILE='DAWAY.DAT',STATUS='NEW')
OPEN(18,FILE='MODEL.DAT',STATUS='NEW')
NUMPTS=0.0
DV=1.0
```

```
C
C*****OBTAIN INITIAL INFORMATION*****
```

```
C
C      WRITE(*,*) 'PROVIDE INITIAL CONDITIONS FOR SPEED AND RPM?'
C      READ(*,*) U0,RPM
C      WRITE(*,*) 'FIRST WAY POINT IN SHIP LENGTHS (XD,YD,ZD)?'
C      READ(*,*) XD,YD,COMZ
C      WRITE(*,*) 'WHAT IS THE DESIRED SPEED?'
C      READ(*,*) UD
C      WRITE(*,*) 'HOW LONG BETWEEN UPDATES FROM THE NAVIGATOR?'
C      READ(*,*) NAVUPDATE
C      WRITE(*,*) ' INPUT TARGET RADIUS IN SHIP LENGTHS.'
C      READ(*,*) TARGET
C      WRITE(*,*) 'INPUT DRIFT VELOCITIES UCO,VCO,WCO IN FT/SEC.'
C      READ(*,*) UCO,VCO,WCO
C      WRITE(*,*) 'INPUT DRIFT VELOCITIES EST.UCE,VCE,WCE IN FT/SEC.'
C      READ(*,*) UCE,VCE,WCE
C      XD1=0.0
C      YD1=0.0
C      XD2=XD
C      YD2=YD
C      DY= (YD2-YD1)
C      DX= (XD2-XD1)
C      ALPHA=ATAN2(DY,DX)
```

```
C
V0 = 0.0
W0 = 0.0
P0 = 0.0
Q0 = 0.0
R0 = 0.0
PHI0 = 0.0
THETA0 = 0.0
PSI0 = 0.0
XPOS0=0.0
```

YPOS0=0.0
ZPOS0=0.0
DB= 0.0
DS = 0.0
DR = 0.0
LATYAW = 0.0
NORPIT = 0.0
RE = U0*L/NU
TNAV=0
XA=XPOS0
YA=YPOS0
LDAWAY = 2.0
NEARMISS = 0.0

C

U = U0
V = V0
W = W0
P = P0
Q = Q0
R = R0
XPOS = XPOS0
YPOS = YPOS0
ZPOS = ZPOS0
PSI = PHIO
THETA = THETA0
PHI = PHIO
QHADOT=0.0
THADOT=0.0
ZHADOT=0.0
QHAT=0.0
THAT=0.0
ZHAT=0.0
VHAT=0.0
ZOBSDOT=0.0
ZZOBS = 0.0

C

C

C

DEFINE LENGTH FRACTIONS FOR GAUSS QUADRATURE TERMS

G4(1) = 0.069431844
G4(2) = 0.330009478
G4(3) = 0.669990521
G4(4) = 0.930568155

C

C

C

DEFINE WEIGHT FRACTIONS FOR GAUSS QUADRATURE TERMS

GK4(1) = 0.1739274225687
GK4(2) = 0.3260725774312
GK4(3) = 0.3260725774312
GK4(4) = 0.1739274225687

C

C

C

DEFINE THE BREADTH BB AND HEIGHT HH TERMS FOR THE INTEGRATION

BR(1) = 75.7/12
BR(2) = 75.7/12
BR(3) = 75.7/12
BR(4) = 55.08/12

C

HH(1) = 16.38/12
HH(2) = 31.85/12
HH(3) = 31.85/12

```

HH(4) = 23.76/12
C
MASS = WEIGHT/G
C
N = 6
DO 15 J = 1,N
  DO 10 K = 1,N
    XMMINV(J,K) = 0.0
    MM(J,K) = 0.0
10  CONTINUE
15  CONTINUE
C
MM(1,1) = MASS - ((RHO/2)*(L**3)*XUDOT)
MM(1,5) = MASS*ZG
MM(1,6) = -MASS*YG
C
MM(2,2) = MASS - ((RHO/2)*(L**3)*YVDOT)
MM(2,4) = -MASS*ZG - ((RHO/2)*(L**4)*YPDOT)
MM(2,6) = MASS*XG - ((RHO/2)*(L**4)*YRDOT)
C
MM(3,3) = MASS - ((RHO/2)*(L**3)*ZWDOT)
MM(3,4) = MASS*YG
MM(3,5) = -MASS*XG - ((RHO/2)*(L**4)*ZQDOT)
C
MM(4,2) = -MASS*ZG - ((RHO/2)*(L**4)*KVDOT)
MM(4,3) = MASS*YG
MM(4,4) = IX - ((RHO/2)*(L**5)*KPDOT)
MM(4,5) = -IXY
MM(4,6) = -IXZ - ((RHO/2)*(L**5)*KRDOT)
C
MM(5,1) = MASS*ZG
MM(5,3) = -MASS*XG - ((RHO/2)*(L**4)*MWDOT)
MM(5,4) = -IXY
MM(5,5) = IY - ((RHO/2)*(L**5)*MQDOT)
MM(5,6) = -IYZ
C
MM(6,1) = -MASS*YG
MM(6,2) = MASS*XG - ((RHO/2)*(L**4)*NVDOT)
MM(6,4) = -IXZ - ((RHO/2)*(L**5)*NPDOT)
MM(6,5) = -IYZ
MM(6,6) = IZ - ((RHO/2)*(L**5)*NRDOT)
LAST=N*N+3*N
DO 20 M=1, LAST
WKAREA(M) = 0.0
20 CONTINUE
C
IER = 0
IA = 6
IDGT = 4
C
*****ROUTINE FOR INVERTING THE MM MATRIX*****
DO 12 I=1,N
  DO 11 J=1,N
    XMMINV(I,J)=0.0
11  CONTINUE
    XMMINV(I,I)=1
12  CONTINUE
CALL INVTA(MM,N,INDX,D)
DO 13 J=1,N
CALL INVTB(MM,N,INDX,XMMINV(1,J))
13 CONTINUE

```

C
C
C
C
C

***** INPUTS *****

RUDDER AND DIVE PLANE COMMANDS

WRITE(*,*) 'INPUT SIMULATION TIME AND TIME STEP'
READ (*,*) SIM,DELT
TIME=0.0
DS= 0.0
DR= 0.0
DB= 0.0
EITA=4.0
BAR=.4

C SIZE OF OUTPUT DATA ARRAY FOR PLOTTING

NUMOUT=6
711 FORMAT (2I4)

C
C

*****SIMULATION BEGINS *****
DO 100 I=1,SIM

C
C
C

PROPULSION MODEL

SIGNU = 1.0
IF (U.LT.0.0) SIGNU = -1.0
IF (ABS(U).LT.X1TEST) U = X1TEST
SIGNN = 1.0
IF (RPM.LT.0.0) SIGNN = -1.0
ETA = 0.012*RPM/U
RE = U*L/NU
CD0 = .00385 + (1.296E-17)*(RE - 1.2E7)**2
CT = ABS(0.008*L**2*ETA*ABS(ETA)/(A0))
CT1 =ABS(0.008*L**2/(A0))
EPS = -1.0+SIGNN/SIGNU*(SQRT(CT+1.0)-1.0)/(SQRT(CT1+1.0)-1.0)
XPROP = CD0*(ETA*ABS(ETA) - 1.0)

C
C
C
C
C

CALCULATE THE DRAG FORCE, INTEGRATE THE DRAG OVER THE VEHICLE
INTEGRATE USING A 4 TERM GAUSS QUADATURE

LATYAW = 0.0
NORPIT = 0.0
DO 500 K = 1,4
UCF(K) = SQRT((V+G4(K)*R*L)**2 + (W-G4(K)*Q*L)**2)
IF(UCF(K).GT.1E-10) THEN
TERM0 = (RHO/2)*(CDY*HH(K)*(V+G4(K)*R*L)**2 +
& CDZ*BR(K)*(W-G4(K)*Q*L)**2)
TERM1 = TERM0*(V+G4(K)*R*L)/UCF(K)
TERM2 = TERM0*(W-G4(K)*Q*L)/UCF(K)
LATYAW = LATYAW + TERM1*GK4(K)*L
NORPIT = NORPIT + TERM2*GK4(K)*L
END IF

500

CONTINUE

C
C
C
C
C

FORCE EQUATIONS

LONGITUDINAL FORCE

FP(1) = MASS*V*R - MASS*W*Q + MASS*XG*Q**2 + MASS*XG*R**2-
& MASS*YG*P*Q - MASS*ZG*P*R + (RHO/2)*L**4*(XPP*P**2 +


```

& XQQ*Q**2 + XRR*R**2 + XPR*P*R) +(RHO/2)*L**3*(XWQ*W*Q +
& XVP*V*P+XVR*V*R+U*Q*(XQDS*DS+XQDB*DB)+XRDR*U*R*DR)+
& (RHO/2)*L**2*(XVV*V**2 + XWW*W**2 + XVDR*U*V*DR + U*W*
& (XWDS*DS+XWDB*DB)+U**2*(XSDS*DS**2+XDBDB*DB**2+
& XDRDR*DR**2))- (WEIGHT -BOY)*SIN(THETA) +(RHO/2)*L**3*
& XQDSN*U*Q*DS*EPS+(RHO/2)*L**2*(XWDSN*U*W*DS+XSDSN*U**2*
& DS**2)*EPS +(RHO/2)*L**2*U**2*XPROP

```

LATERAL FORCE

```

FP(2) = -MASS*U*R - MASS*XG*P*Q + MASS*YG*R**2 - MASS*ZG*Q*R +
& (RHO/2)*L**4*(YPQ*P*Q + YQR*Q*R)+(RHO/2)*L**3*(YP*U*P +
& YR*U*R + YVQ*V*Q + YWP*W*P + YWR*W*R) + (RHO/2)*L**2*
& (YV*U*V + YVW*V*W + YDR*U**2*DR) -LATYAW +(WEIGHT-BOY)*
& COS(THETA)*SIN(PHI)+MASS*W*P+MASS*YG*P**2

```

NORMAL FORCE

```

FP(3) = MASS*U*Q - MASS*V*P - MASS*XG*P*R - MASS*YG*Q*R +
& MASS*ZG*P**2 + MASS*ZG*Q**2 + (RHO/2)*L**4*(ZPP*P**2 +
& ZPR*P*R + ZRR*R**2) + (RHO/2)*L**3*(ZQ*U*Q + ZVP*V*P +
& ZVR*V*R) +(RHO/2)*L**2*(ZW*U*W + ZVV*V**2 + U**2*(ZDS*
& DS+ZDB*DB))-NORPIT+(WEIGHT-BOY)*COS(THETA)*COS(PHI)+
& (RHO/2)*L**3*ZQN*U*Q*EPS +(RHO/2)*L**2*(ZWN*U*W + ZDSN*
& U**2*DS)*EPS

```

ROLL FORCE

```

FP(4) = -IZ*Q*R +IY*Q*R -IXY*P*R +IYZ*Q**2 -IYZ*R**2 +IXZ*P*Q +
& MASS*YG*U*Q -MASS*YG*V*P -MASS*ZG*W*P+(RHO/2)*L**5*(KPQ*
& P*Q + KQR*Q*R) +(RHO/2)*L**4*(KP*U*P +KR*U*R + KVQ*V*Q +
& KWP*W*P + KWR*W*R) +(RHO/2)*L**3*(KV*U*V + KVW*V*W) +
& (YG*WEIGHT - YB*BOY)*COS(THETA)*COS(PHI) - (ZG*WEIGHT -
& ZB*BOY)*COS(THETA)*SIN(PHI) + (RHO/2)*L**4*KPN*U*P*EPS+
& (RHO/2)*L**3*U**2*KPROP +MASS*ZG*U*R

```

PITCH FORCE

```

FP(5) = -IX*P*R +IZ*P*R +IXY*Q*R -IYZ*P*Q -IXZ*P**2 +IXZ*R**2 -
& MASS*XG*U*Q + MASS*XG*V*P + MASS*ZG*V*R - MASS*ZG*W*Q +
& (RHO/2)*L**5*(MPP*P**2 +MPR*P*R +MRR*R**2)+(RHO/2)*L**4*
& (MQ*U*Q + MVP*V*P + MVR*V*R) + (RHO/2)*L**3*(MW*U*W +
& MVV*V**2+U**2*(MDS*DS+MDB*DB))+ NORPIT -(XG*WEIGHT-
& XB*BOY)*COS(THETA)*COS(PHI)+(RHO/2)*L**4*MQN*U*Q*EPS +
& (RHO/2)*L**3*(MWN*U*W+MDSN*U**2*DS)*EPS-
& (ZG*WEIGHT-ZB*BOY)*SIN(THETA)

```

YAW FORCE

```

FP(6) = -IY*P*Q +IX*P*Q +IXY*P**2 -IXY*Q**2 +IYZ*P*R -IXZ*Q*R -
& MASS*XG*U*R + MASS*XG*W*P - MASS*YG*V*R + MASS*YG*W*Q +
& (RHO/2)*L**5*(NPQ*P*Q + NQR*Q*R) +(RHO/2)*L**4*(NP*U*P+
& NR*U*R + NVQ*V*Q +NWP*W*P + NWR*W*R) +(RHO/2)*L**3*(NV*
& U*V + NVW*V*W + NDR*U**2*DR) - LATYAW + (XG*WEIGHT -
& XB*BOY)*COS(THETA)*SIN(PHI)+(YG*WEIGHT)*SIN(THETA)
& +(RHO/2)*L**3*U**2*NPROP-YB*BOY*SIN(THETA)

```

NOW COMPUTE THE F(1-6) FUNCTIONS

```

DO 600 J = 1,6
      F(J) = 0.0
DO 600 K = 1,6
      F(J) = XMMINV(J,K)*FP(K) + F(J)
600  CONTINUE
C
C THE LAST SIX EQUATIONS COME FROM THE KINEMATIC RELATIONS
C
C FIRST SET THE DRIFT CURRENT VALUES
C
C   UCO = 0.0
C   VCO = 0.0
C   WCO = 0.0
C
C INERTIAL POSITION RATES F(7-9)
C
C F(7) = UCO + U*COS(PSI)*COS(THETA) + V*(COS(PSI)*SIN(THETA)*
& SIN(PHI) - SIN(PSI)*COS(PHI)) + W*(COS(PSI)*SIN(THETA)*
& COS(PHI) + SIN(PSI)*SIN(PHI))
C
C F(8) = VCO + U*SIN(PSI)*COS(THETA) + V*(SIN(PSI)*SIN(THETA)*
& SIN(PHI) + COS(PSI)*COS(PHI)) + W*(SIN(PSI)*SIN(THETA)*
& COS(PHI) - COS(PSI)*SIN(PHI))
C
C F(9) = WCO - U*SIN(THETA) + V*COS(THETA)*SIN(PHI) + W*COS(THETA)*
& COS(PHI)
C
C EULER ANGLE RATES F(10-12)
C
C F(10) = P + Q*SIN(PHI)*TAN(THETA) + R*COS(PHI)*TAN(THETA)
C
C F(11) = Q*COS(PHI) - R*SIN(PHI)
C
C F(12) = Q*SIN(PHI)/COS(THETA) + R*COS(PHI)/COS(THETA)
C
C UDOT = F(1)
C VDOT = F(2)
C WDOT = F(3)
C PDOT = F(4)
C QDOT = F(5)
C RDOT = F(6)
C XDOT = F(7)
C YDOT = F(8)
C ZDOT = F(9)
C PHIDOT = F(10)
C THETAD = F(11)
C PSIDOT = F(12)
C
C ***** CREATE OUTPUT DATA FILE *****
C
C IF (I .EQ. DV ) THEN
C   TIMER=FLOAT(I)/2.
C   WRITE (20,*) I
C   WRITE (20,744) DAWAY
C   WRITE (18,*) I
C   WRITE (18,743) DS/.01745 ,DR/.01745
C   WRITE (18,744) XPOS/L,YPOS/L,ZPOS/L,XD,YD,COMZ,DAWAY
C   WRITE (18,745) U,UD,RPM
743  FORMAT (2E11.3)
744  FORMAT (6E12.4)

```



```

WRITE(*,*) 'WHAT IS THE NEXT WAY POINT (XD,YD,ZD)?'
READ (*,*) XD,YD,COMZ
XD1=XD2
YD1=YD2
XD2=XD
YD2=YD
DY= (YD2-YD1)
DX= (XD2-XD1)
ALPHA=ATAN2(DY,DX)
WRITE (*,*) 'WHAT IS THE DESIRED SPEED?'
READ (*,*) UD
ENDIF

```

```

***** NAVIGATOR *****

```

```

IF ((TIME-TNAV) .GE. NAVUPDATE ) THEN
XA=XPOS/L
YA=YPOS/L
TNAV=TNAV+NAVUPDATE
ENDIF

```

```

***** HEADING *****

```

```

IF(ABS(XD-XA).LE.0.00000001)THEN
  IF((YD-YA).GT.0.0) HD = 2.*ATAN(1.)
  IF((YD-YA).LT.0.0) HD = - 2.*ATAN(1.)
ELSEIF ((XD-XA).LT.0.0 .AND. (YD-YA) .GT. 0.0) THEN
  HD = ATAN((YD-YA)/(XD-XA)) + 4.*ATAN(1.)
ELSEIF ((XD-XA).LT.0.0 .AND. (YD-YA) .LT. 0.0) THEN
  HD = ATAN((YD-YA)/(XD-XA)) - 4.*ATAN(1.)
ELSE
  HD = ATAN((YD-YA)/(XD-XA))
ENDIF
HDMDEG = HDM*360./(8.*ATAN(1.))

```

```

***** SWAY VELOCITY OBSERVER *****

```

```

VHAT = 0.0

```

```

THE VALUE OF THE COEFFICIENTS VARY WITH SPEED TO ACCOUNT FOR DIFFERENCES
BETWEEN THIS NONLINEAR MODEL AND THE LINEARIZED MODEL THE OBSERVER IS
BASED ON.(THIS CODE WAS NOT USED, INSTEAD VHAT WAS SET EQUAL TO ZERO.)

```

```

FF = -2.0
LLL= -2.275*(U**6)+48.102*(U**5)-406.839*(U**4)+
& 1763.181*(U**3)-4165.344*(U**2)+5265.448*U-3211.956
GG = 1.1329*U - 70.535
HHH = .07584*(U**2) - 3.0509*U + .0003
ZZOBSDOT = HHH*DR + GG*R + FF*VHAT
ZZOBS = ZZOBS + ZZOBSDOT*DELT
VHAT = ZZOBS + LLL*R

```

```

***** CONDITIONS TO ALLOW SHORTEST TURN *****

```

```

HDP = HD + 8.*ATAN(1.)
IF (ABS(HDP-PSI) .LT. ABS(HD-PSI)) THEN
  HDM = 8*ATAN(1.) + HD
ELSE
  HDM=HD
ENDIF

```

```

C      ***** BEGIN HEADING SMC CALCULATIONS *****
C
VCC=VCE*COS(ALPHA) -UCE*SIN(ALPHA)
SS2=.4767*(PSI-HDM) + .0121*VHAT + .8790*R + .4767*ASIN(VCC/U)
IF(ABS(SS2) .LT. .1) SATSGN2=(SS2/.10)
IF(SS2 .LE. -.1) SATSGN2=-1.0
IF(SS2 .GE. .1) SATSGN2=1.0
C
DR= 3.1007*SATSGN2 + .1373*VHAT + .5081*R
IF (DR .GE. 0.4) DR = 0.4
IF (DR .LE. - 0.4) DR = -0.4
C
***** RPM INPUT CALCULATION *****
SS1=U-UD
IF(ABS(SS1) .LT. 1.0) SATSGN1=(SS1/1)
IF(SS1 .LE. -1.0) SATSGN1=-1.0
IF(SS1 .GE. 1.0) SATSGN1=1.0
RPM=-4000.0*SATSGN1 + 83.33*U
IF (RPM .GE. 5000.0) RPM= 5000.0
IF (RPM .LE. -5000.0) RPM=-5000.0
C
TIME=TIME+DELT
C
PHIANG = PHI/0.0174532925
THEANG = THETA/0.0174532925
PSIANG = PSI/0.0174532925
C
TRAC=-YPOS
ROLL=PHIANG
YAW=PSIANG
DEPTH=-ZPOS
PITCH=THEANG
BOWANG=(DB/.01745)
STNANG=(DS/.01745)
100 CONTINUE
WRITE(*,*) 'NPTS = ',NUMPTS
WRITE(*,*) 'TIMEINTERVAL = ',DELT
WRITE(*,*) 'NAVIGATOR UPDATE TIME = ',NAVUPDATE
WRITE(*,*) 'TARGET RADIUS = ',TARGET
WRITE(*,*) 'NONDIM SPEED = ',(UCO**2 +VCO**2)**.5/4
WRITE(*,*) 'NEARMISS = ',NEARMISS
STOP
END
C
***** DEPTH CONTROL OBSERVER *****
C
SUBROUTINE OBSER(QHADOT,THADOT,ZHADOT,QHAT,THAT,ZHAT,DELT,ZPOS,
* DS,U)
C
QHADOT=-0.7*QHAT-0.03*THAT-0.035*DS-20.9293*(ZPOS-ZHAT)
THADOT=QHAT-14.4092*(ZPOS-ZHAT)
ZHADOT=-6*THAT+16.45*(ZPOS-ZHAT)
C
QHAT= QHAT+DELT*QHADOT
THAT= THAT+DELT*THADOT
ZHAT= ZHAT+DELT*ZHADOT
RETURN
END

```

APPENDIX C.

C
C
C
C
C

```
*****
* THIS PROGRAM PLOTS THE OUTPUT GRAPHS FOR THE SIMULATION OF THE *
* NONLINEAR MODEL OF THE SDV-9 SIMULATION RUNS. *
*****
```

```
REAL*4 T(600),DS(600),DR(600),OHS(600)
REAL*4 XPOS(600),YPOS(600),ZPOS(600)
REAL*4 ROLL(600),PITCH(600),YAW(600)
REAL*4 RPM(600),U(600)
REAL*4 UG(600),HG(600),UD(600),XD(600),YD(600),COMZ(600)
REAL*4 DEPTH(600),DEPTHD(600),DAWAY(600)
CHARACTER*1 ANS
WRITE(*,*) 'INPUT NPTS & THE TIME INCREMENT (DELT)'.
READ(*,*) NPTS,DELT
DO I=1,NPTS
OPEN(18,FILE='MODEL.DAT',STATUS='OLD')
READ(18,*) INC
T(I) = DELT*INC
READ(18,*) DS(I),DR(I)
READ(18,*) XPOS(I),YPOS(I),ZPOS(I),XD(I),YD(I),COMZ(I),DAWAY(I)
READ(18,*) U(I),UD(I),RPM(I)
OHS(I)=0.0
DEPTH(I) = -ZPOS(I)
DEPTHD(I) = -COMZ(I)
ENDDO
```

```
CALL MXMN(T,NPTS,TMX,TMN)
CALL MXMN(DS,NPTS,DSMX,DSMN)
CALL MXMN(DR,NPTS,DRMX,DRMN)
CALL MXMN(XPOS,NPTS,XPOSMX,XPOSMN)
CALL MXMN(YPOS,NPTS,YPOSMX,YPOSMN)
CALL MXMN(DEPTH,NPTS,DEPTHMX,DEPTHMN)
CALL MXMN(DEPTHD,NPTS,DEPTHDMX,DEPTHDMN)
CALL MXMN(U,NPTS,UMX,UMN)
CALL MXMN(UD,NPTS,UDMX,UDMN)
CALL MXMN(RPM,NPTS,RPMMX,RPMMN)
CALL MXMN(XD,NPTS,XDMX,XDMN)
CALL MXMN(YD,NPTS,YDMX,YDMN)
CALL MXMN(DAWAY,NPTS,DAWAYMX,DAWAYMN)
```

```
CALL SCALE(TMN,TMX,4.1,TORIG,TSTP,TMAX)
```

```
IF(DSMX-DSMN .EQ. 0.0) THEN
DSORIG=-.1
DSSTP=.1
DSMAX=.1
ELSE
CALL SCALE(DSMN,DSMX,4.1,DSORIG,DSSTP,DSMAX)
ENDIF
```

```
IF(DRMX-DRMN .EQ. 0.0) THEN
DRORIG=-.1
DRSTP=.1
DRMAX=.1
ELSE
```

C
C
C

```

CALL SCALE(DRMN,DRMX,4.1,DRORIG,DRSTP,DRMAX)
ENDIF
C
IF(XPOSMX-XPOSMN .EQ. 0.0) THEN
XPOSORIG=-.1
XPOSSTP=.1
XPOSMAX=.1
ELSE
CALL SCALE(XPOSMN,XPOSMX,4.1,XPOSORIG,XPOSSTP,XPOSMAX)
ENDIF
C
XXMX=AMAX1(XPOSMX,XDMX)
XXMN=AMIN1(XPOSMN,XDMN)
IF(XXMX-XXMN .EQ. 0.0) THEN
XXORIG=-.1
XXSTP=.1
XXMAX=.1
ELSE
CALL SCALE(XXMN,XXMX,6.1,XXORIG,XXSTP,XXMAX)
ENDIF
C
YYMX=AMAX1(YPOSMX,YDMX)
YYMN=AMIN1(YPOSMN,YDMN)
IF(YYMX-YYMN .EQ. 0.0) THEN
YYORIG=-.1
YYSTP=.1
YYMAX=.1
ELSE
CALL SCALE(YYMN,YYMX,6.1,YYORIG,YYSTP,YYMAX)
ENDIF
C
DDMX=AMAX1(DEPTHMX,DEPTHDMX)
DDMN=AMIN1(DEPTHMN,DEPTHDMN)
IF(DDMX-DDMN .EQ. 0.0) THEN
DDORIG=-.1
DDSTP=.1
DDMAX=.1
ELSE
CALL SCALE(DDMN,DDMX,6.1,DDORIG,DDSTP,DDMAX)
ENDIF
C
IF(DAWAYMX-DAWAYMN .EQ. 0.0) THEN
DAWAYORIG=-.1
DAWAYSTP=.1
DAWAYMAX=.1
ELSE
CALL SCALE(DAWAYMN,DAWAYMX,6.1,DAWAYORIG,DAWAYSTP,DAWAYMAX)
ENDIF
C
HGMX=AMAX1(PSIMX,HDMX)
HGMN=AMIN1(PSIMN,HDMN)
IF(HGMX-HGMN .EQ. 0.0) THEN
HGORIG=-.1
HGSTP=.1
HGMAX=.1
ELSE
CALL SCALE(HGMN,HGMX,6.1,HGORIG,HGSTP,HGMAX)
ENDIF
C
UGMX=AMAX1(UMX,UDMX)

```

```

UGMN=AMIN1(UMN,UDMN)
IF(UGMX-UGMN .EQ. 0.0) THEN
UGORIG=-.1
UGSTP=.1
UGMAX=.1
ELSE
CALL SCALE(UGMN,UGMX,6.1,UGORIG,UGSTP,UGMAX)
ENDIF

C
IF(RPMMX-RPMMN .EQ. 0.0) THEN
RPMORIG=-.1
RPMSTP=.1
RPMMAX=.1
ELSE
CALL SCALE(RPMMN,RPMMX,4.1,RPMORIG,RPMSTP,RPMMAX)
ENDIF

C***** PLOT OR VIEW OPTION *****
C
C
4  WRITE(*,*) 'DO YOU WANT TO VIEW THE PLOT OR OBTAIN A HARDCOPY?'
WRITE(*,*) ' '
WRITE(*,*) 'INPUT 1 FOR VIEW OR 2 FOR HARDCOPY'
READ(*,*) IPLOT_VAL
IF(IPLOT_VAL .EQ. 1) CALL PGPX
IF(IPLOT_VAL .EQ. 2) CALL LN03I

C
C
C ***** HORIZONTAL POSITION *****
CALL PAGE(8.5,11.0)
CALL NOBRDR
CALL PHYSOR(1.75,7.75)
CALL AREA2D(2.0,2.0)
CALL XTICKS(1)
CALL YTICKS(1)
CALL XNAME('X (SHIP LENGTHS)$',100)
CALL YNAME('Y (SHIP LENGTHS)$',100)
CALL GRAF(XXORIG,XXSTP,XXMAX,YYORIG,YYSTP,YYMAX)
IMARK=0
CALL RESET('DOT')
CALL RESET('THKCRV')
CALL GRID(1,1)
CALL THKCRV(4)
CALL CURVE(XPOS,YPOS,NPTS,IMARK)
CALL MARKER(13)
CALL SCLPIC(2)
IMARK=-1
CALL CURVE(XD,YD,NPTS,IMARK)
CALL ENDGR(0)

C
C ***** STERN PLANE INPUTS *****
C
C
C CALL PHYSOR(4.95,5.05)
C CALL AREA2D(2.0,2.0)
C CALL XTICKS(1)
C CALL YTICKS(1)
C CALL XNAME('TIME (SEC)$',100)
C CALL YNAME('STERN PL. (DEGREES)$',100)
C CALL GRAF(TORIG,TSTP,TMAX,DSORIG,DSSTP,DSMAX)
C IMARK=0
C CALL RESET('DOT')

```


C
C
C
C
C
C
C
C

```
CALL RESET('THKCRV')
CALL GRID(1,1)
CALL THKCRV(4)
CALL CURVE(T,DS,NPTS,IMARK)
CALL ENDGR(0)
```

***** RUDDER INPUTS *****

```
CALL PHYSOR(4.95,7.75)
CALL AREA2D(2.,2.)
CALL XTICKS(1)
CALL YTICKS(1)
CALL YINTAX
CALL XNAME('TIME (SEC)$',100)
CALL YNAME('RUDDER (DEGREES)$',100)
CALL GRAF(TORIG,TSTP,TMAX,DRORIG,DRSTP,DRMAX)
CALL RESET('DOT')
CALL RESET('THKCRV')
CALL GRID(1,1)
IMARK=0
CALL THKCRV(4)
CALL CURVE(T,DR,NPTS,IMARK)
CALL ENDGR(0)
```

C
C
C

***** DISTANCE AWAY FROM NEXT WAY POINT (DAWAY) *****

```
CALL PHYSOR(1.75,5.05)
CALL AREA2D(2.,2.)
CALL XTICKS(1)
CALL YTICKS(1)
CALL XNAME('TIME (SEC)$',100)
CALL YNAME('DAWAY (SHIP LENGTHS) $',100)
CALL GRAF(TORIG,TSTP,TMAX,DAWAYORIG,DAWAYSTP,DAWAYMAX)
CALL RESET('DOT')
CALL RESET('THKCRV')
CALL GRID(1,1)
IMARK=0
CALL THKCRV(4)
CALL CURVE(T,DAWAY,NPTS,IMARK)
CALL ENDGR(0)
```

C
C
C

***** SPEED VS DESIRED SPEED *****

```
CALL PHYSOR(4.95,5.05)
CALL AREA2D(2.,2.)
CALL XTICKS(1)
CALL YTICKS(1)
CALL XNAME('TIME (SEC)$',100)
CALL YNAME('U AND UD (FT/SEC) $',100)
CALL GRAF(TORIG,TSTP,TMAX,UGORIG,UGSTP,UGMAX)
CALL RESET('DOT')
CALL RESET('THKCRV')
CALL GRID(1,1)
IMARK=0
CALL THKCRV(4)
CALL CURVE(T,U,NPTS,IMARK)
CALL DOT
CALL CURVE(T,UD,NPTS,IMARK)
CALL ENDGR(0)
```

C

```

C ***** MOTOR INPUT *****
C
C CALL PHYSOR(4.95,2.35)
C CALL AREA2D(2.,2.)
C CALL XTICKS(1)
C CALL YTICKS(1)
C CALL XNAME('TIME (SEC)$',100)
C CALL YNAME('RPM $',100)
C CALL YINTAX
C CALL GRAF(TORIG,TSTP,TMAX,RPMORIG,RPMSTP,RPMMAX)
C CALL RESET('DOT')
C CALL RESET('THKCRV')
C CALL GRID(1,1)
C IMARK=0
C CALL THKCRV(2)
C CALL CURVE(T,RPM,NPTS,IMARK)
C CALL ENDPL(0)
C GOTO 100
C
100 CONTINUE
C CALL METAF(1)
C CALL ENDPL(0)
C CALL DONEPL
C WRITE (*,*) ' DO YOU WANT TO CONTINUE ? (Y/N)'
C READ(*,101) ANS
C IF (ANS .EQ. 'Y') GOTO 4
C STOP
101 FORMAT(A)
C END

```

APPENDIX D.

The values used to design a reduced order observer for sway velocity(v) varied so much with respect to changes in surge velocity(u) that the following attempt was made to relate them using curve fitting techniques. The following is the computer code which resulted.

```
***** SWAY VELOCITY OBSERVER *****  
VHAT = 0.0  
  
THE VALUE OF THE COEFFICIENTS VARY WITH SPEED TO ACCOUNT FOR DIFFERENCES  
BETWEEN THIS NONLINEAR MODEL AND THE LINEARIZED MODEL THE OBSERVER IS  
BASED ON.(THIS CODE WAS NOT USED, INSTEAD VHAT WAS SET EQUAL TO ZERO.)  
  
FF = -2.0  
LLL= -2.275*(U**6)+48.102*(U**5)-406.839*(U**4)+  
& 1763.181*(U**3)-4165.344*(U**2)+5265.448*U-3211.956  
GG = 1.1329*U - 70.535  
HHH = .07584*(U**2) - 3.0509*U + .0003  
ZZOBSDOT = HHH*DR + GG*R + FF*VHAT  
ZZOBS = ZZOBS + ZZOBSDOT*DELT  
VHAT = ZZOBS + LLL*R
```

FIGURE 48. Adaptive Sway Velocity Observer Code

The coefficients in the above equations and the quality of the curve fitting technique that provided them are shown in the following graphs. Notice the wide range of values that occur when dealing with the full nonlinear model. This again shows the extent to which the sliding mode control laws must deal with unmodeled behavior.

Recall that ultimately sway velocity was set equal to zero and treated as a disturbance.

DATA

Point #1:	X=.5	Y=-1425
Point #2:	X=1	Y=-704.97
Point #3:	X=1.5	Y=-464.925
Point #4:	X=2	Y=-344.901
Point #5:	X=3	Y=-224.877
Point #6:	X=4	Y=-164.866
Point #7:	X=5	Y=-128.859
Point #8:	X=6	Y=-104.854

POLYNOMIAL MODEL: $Y=A(M)*X^M+A(M-1)*X^{(M-1)}+...+A(1)*X+A(0)$
Coefficients:
A(0)=-3211.95616
A(1)=5265.4479
A(2)=-4165.344
A(3)=1763.18129
A(4)=-406.83942
A(5)=48.102117
A(6)=-2.2753449

Source	Df	SS	MS	F
Regression	6	1380885.259	230147.543	2607.572
Residual	1	88.261	88.261	
Total	7	1380973.521		

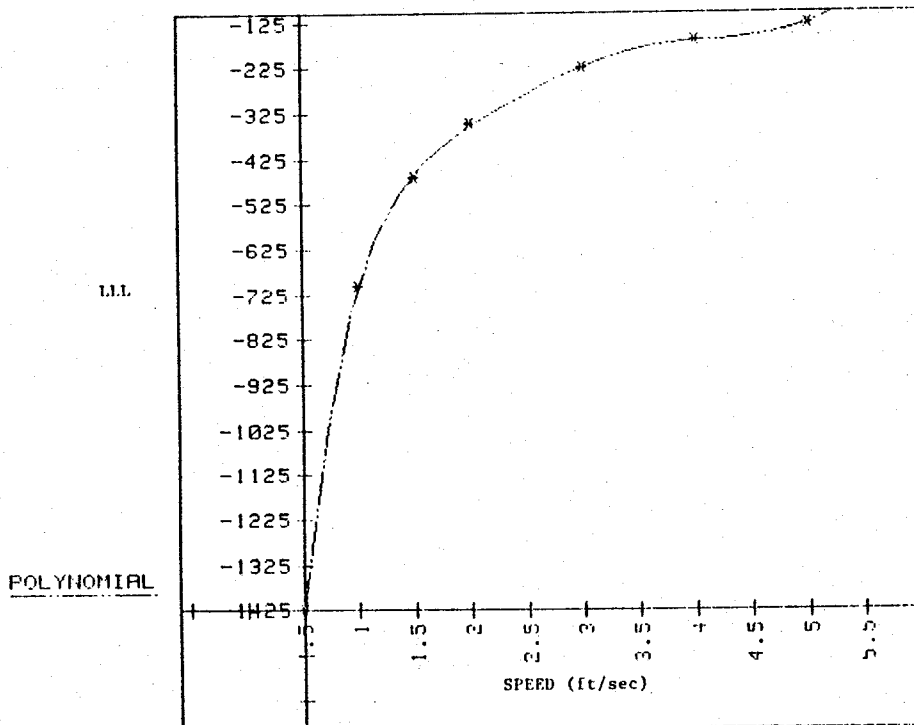


Figure 49. Graph of LLL vs Time

DATA

Point #1:	X=.5	Y=-1.506
Point #2:	X=1	Y=-2.975
Point #3:	X=1.5	Y=-4.405
Point #4:	X=2	Y=-5.798
Point #5:	X=3	Y=-8.47
Point #6:	X=4	Y=-10.989
Point #7:	X=5	Y=-13.358
Point #8:	X=6	Y=-15.575

POLYNOMIAL MODEL: $Y=A(0)*X^0+A(1)*X^1+A(2)*X^2$
 Coefficients:
 $A(0)=-.000121149$
 $A(1)=-3.050596472$
 $A(2)=-.0757941142$

Source	Df	SS	MS	F
Regression	2	180.333	90.166	
518197060.718 H4D.3D				
Residual	5	.000	.000	
Total	7	180.333		

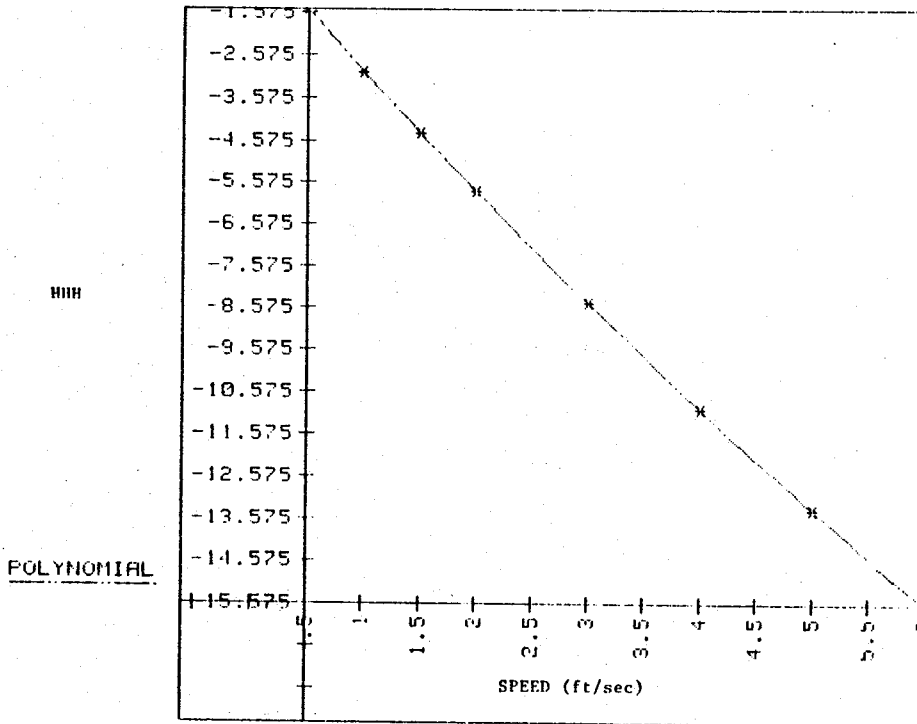


Figure 50. Graph of HHH vs Speed

DATA

Point #1:	X=.5	Y=-69.97
Point #2:	X=1	Y=-69.403
Point #3:	X=1.5	Y=-68.836
Point #4:	X=2	Y=-68.269
Point #5:	X=3	Y=-67.136
Point #6:	X=4	Y=-66.002
Point #7:	X=5	Y=-64.869
Point #8:	X=6	Y=-63.735

POLYNOMIAL MODEL: $Y=A(N)*X^N+A(N-1)*X^{(N-1)}+...+A(1)*X+A(0)$
 Coefficients:
 A(0)=-70.536681722
 A(1)=1.133728049
 A(2)=-.000024506

Source	Df	SS	MS	F
Regression	2	35.176	17.588	
160213160.189 M4D.3D				
Residual	5	.000	.000	
Total	7	35.176		

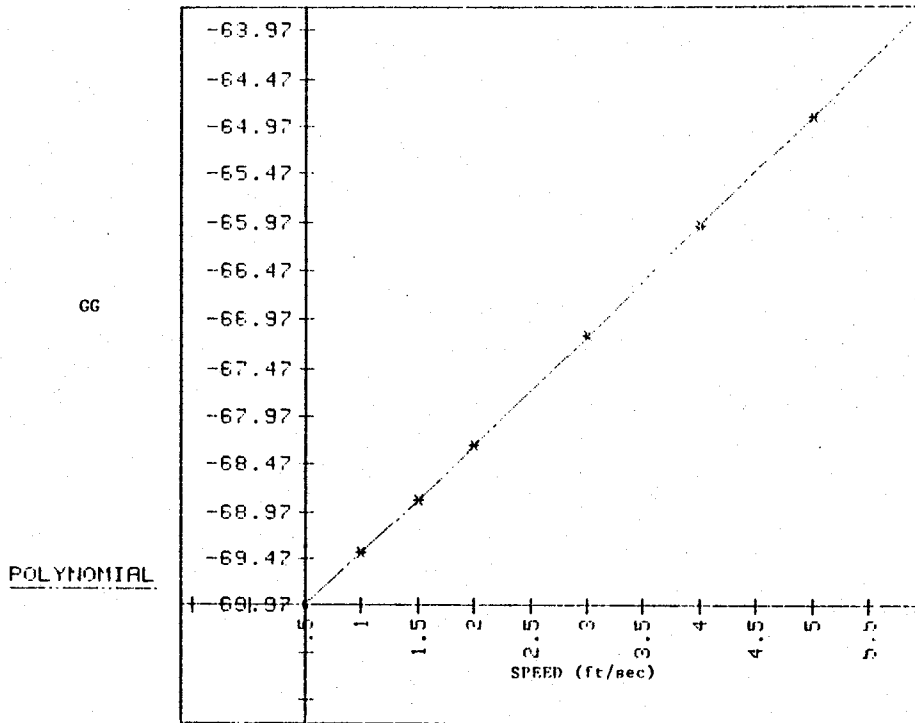


Figure 51. Graph of GG vs Speed

LIST OF REFERENCES

1. Joo-No Sur *Design and Investigation of a Dive Plane Sliding Mode Compensator for an Autonomous Underwater Vehicle* Master's Thesis, Naval Postgraduate School, Monterey, California, September 1989.
2. Richard J. Boncal *A Study of Model Based Maneuvering Controls for Autonomous Underwater Vehicles* Master's Thesis, Naval Postgraduate School, Monterey, California, December 1987.
3. NSCS Technical Memorandum 231-78, *SDV Simulator Hydrodynamic Coefficients*, by N. S. Smith, J. W. Crane, and D. C. Summey, June 1978.
4. Raymond A. Decarlo, Stainislaw H. Zak, and Gregory P. Matthews *Variable Structure Control of Nonlinear Multivariable Systems* Proceeding of the IEEE, VOL. 76, NO.3, March 1988.
5. Dana R. Yoerger and Jean-Jacques E. Slotine *Robust Trajectory Control of Underwater Vehicles* IEEE Journal of Oceanic Engineering, VOL. OE-10, NO 4, October 1985.
6. Katsuhiko, Ogata *Modern Control Engineering* Prentice-Hall Electrical Engineering Series, Prentice-Hall, Inc., Engelwood Cliffs, N. J., 1970.
7. F. A. Papoulias, R. Christi, D. Marco, and A. J. Healey *Modeling, Sliding Mode Control Design, and Visual Simulation of AUV Dive Plane Dynamics Response* Proceeding, 6th International Symposium on Unmanned Untethered Submersible Technology, Washington D. C., June 1989.
8. Bernard Friendland *Control System Design* Mc Graw-Hill Book Company, 1986.
9. Vadim I. Utkin *Variable Structure System with Sliding Mode* IEEE Transactions on Automatic Control, April 1977.

10. Gordon S. MacDonald *Model Based Design and Verification of Rapid Dive Controller for Autonomous Underwater Vehicles* Master's Thesis Naval Postgraduate School, Monterey, California, March 1989.
11. Healey A. J., Papoulias, F. A. and Lienard D. E. *Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles* Proceedings, International Conference on Modelling and Control of Marine Craft, Exeter, U. K., April 20, 1990.

INITIAL DISTRIBUTION LIST

	<u>Copies</u>
1. Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2. Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3. Chairman, Code MEHy Department of Mechanical Engineering Naval Postgraduate School Monterey, CA 93943-5000	8
4. Dr. G. Dobeck, Code 4210 Naval Coastal Systems Command Panama City, FL 32407-5000	1
5. Hal Cook, Code U25 Naval Surface Weapons Center White Oak, MD 20910	1
6. Paul Heckman, Code 943 Head, Undersea AI & Robotics Branch Naval Ocean Systems Center San Diego, CA 92152	1
7. Robert Wilson Head, Systems Engineering Branch DTRC, Carderock Bethesda, MD 20084-5000	1
8. RADM Evans, Code C92R Naval Sea Systems Command Washington, DC 20362	1

- | | | |
|-----|--|---|
| 9. | <p>Dan Steiger
 Marine Systems Group
 Naval Research Laboratory
 Washington, DC 20032</p> | 1 |
| 10. | <p>Dick Blidberg
 Marine Systems Engineering Lab
 SERB Building 242
 University of New Hampshire
 Durham, NH 03824</p> | 1 |
| 11. | <p>Jennifer Rau, Code U25
 Naval Surface Weapons Center
 Silver Spring, MD 20903-5000</p> | 1 |
| 12. | <p>Technical Library Branch, Code E23
 Naval Surface Warfare Center
 Silver Spring, MD 20903-5000</p> | 1 |
| 13. | <p>Naval Engineering Curricular Office, Code 34
 Naval Postgraduate School
 Monterey, CA 93943-5000</p> | 1 |
| 14. | <p>LCDR David E. Lienard
 Supervisor of Shipbuilding Office
 Newport News Shipbuilding Co.
 Newport News, VA 23607</p> | 1 |
| 15. | <p>Mr. Glenn Reid, Code U401
 Naval Surface Warfare Center
 Silver Spring, MD 20901</p> | 1 |
| 16. | <p>Dr. Dana Yoeger
 Woods Hole Oceanographic Institute
 Woods Hole, MA 02543</p> | 1 |
| 17. | <p>Professor F. A. Papoulias, Code MEPa
 Department of Mechanical Engineering
 Naval Postgraduate School
 Monterey, CA. 93943-5000</p> | 1 |

18. Professor R. McGhee, Code 52Mz 1
Department of Computer Science
Naval Postgraduate School
Monterey, CA. 93943-5000
19. Professor R. Christi, Code 62Cx 1
Department of Electrical & Computer Engineering
Naval Postgraduate School
Monterey, CA. 93943-5000
20. Sur Joo-No 1
Naval Academy, Jinhae, Ggungnam 602-02
Republic of Korea

